Interviews with participants

Interviews with participants ***1***

Participant 1 .... **2**
  Beginning of the interview with participant 1 2
  Participant-Respondent Interview (ChatGPT) 3
  Conclusion of the interview with participant 1 13

Participant 2 .... **15**
  Beginning of the interview with participant 2 15
  Participant-Respondent Interview (ChatGPT) 16
  Conclusion of the interview with participant 2 25

Participant 3 .... **27**
  Beginning of the interview with participant 3 27
  Participant-Respondent Interview (ChatGPT) 28
  Conclusion of the interview with participant 3 37

Participant 4 .... **39**
  Beginning of the interview with participant 4 39
  Participant-Respondent Interview (ChatGPT) 40
  Conclusion of the interview with participant 4 53

Participant 5 .... **55**
  Beginning of the interview with participant 5 55
  Participant-Respondent Interview (ChatGPT) 56
  Conclusion of the interview with participant 5 68

Participant 6 .... **69**
  Beginning of the interview with participant 6 69
  Participant-Respondent Interview (ChatGPT) 70
  Conclusion of the interview with participant 6 77

Participant 7 .... **79**
  Beginning of the interview with participant 7 79
  Participant-Respondent Interview (ChatGPT) 80
  Conclusion of the interview with participant 7 91

Participant 8 .... **93**
  Beginning of the interview with participant 8 93
  Participant-Respondent Interview (ChatGPT) 94
  Conclusion of the interview with participant 8 104

Participant 1

Beginning of the interview with participant 1

Interviewer: Hello, thank you for taking the time to talk to us. Before we start with the work questions – how are you?

Respondent: Hello, thanks for asking. I'm doing quite well, although it's quite a busy time right now - a lot of projects are running in parallel, but that's kind of part of this job. And how about you?

Interviewer: I'm fine too, thank you. So let's get straight to the point - how long have you actually been working as a business analyst?

Respondent: I've been working as a business analyst for twelve years. I started right after school, although at first it was more of a combined role with project management, so it was a classic "a bit of everything", but gradually I started to specialize in analysis.

Interviewer: And how did you get the role? Was it a deliberate choice, or did it come about through circumstances?

Respondent: Well, it wasn't exactly the plan from the beginning. After college, I went freelance, working for smaller companies, where it was useful if you could handle both analysis and, perhaps, solution design or communication with the customer. I found that I enjoyed understanding needs and translating them into specific requirements the most. So I started to educate myself more in the field of business analysis and gained experience directly in that role.

Interviewer: And did you have any other position before you started as an analyst?

Respondent: Yeah, I worked as a project coordinator for a while. It was a smaller IT company where I led smaller teams, planned tasks, communicated with customers. That's where I actually realized that I enjoyed being between IT and business, finding compromises and solving what the customer really wanted – and that led me to analysis.

Interviewer: And in what types of projects do you have the most experience? IT, process changes, strategy...?

Respondent: I worked mostly on IT projects, but they were often projects with an overlap into process analysis – for example, implementing a new system that was supposed to automate or streamline existing procedures. I did strategic projects more on the side, for example when we were working on a long-term digitalization plan with a client. But my daily bread is more of the "operational" things – collecting requirements, analyzing processes, proposing changes, communicating with developers.

Interviewer: I understand. And as for the environment – did you work in different companies, or rather in one for a long time?

Respondent: Definitely in different ones. I worked as a self-employed person, so I went through nine companies in those twelve years. Each one was a little different – some smaller, agile startups, others corporate environments with rigid processes. It was a great school because I had the opportunity to get to know different approaches, cultures and types of projects.

Interviewer: And where do you work now? Is it a larger or smaller company?

Respondent: I am now in one of the largest companies I have ever worked for – over a thousand employees. It is a big change compared to smaller companies, especially in terms of structure and processes. Everything takes longer, but there is also the opportunity to specialize, to delve deeper into more complex topics. It has its pros and cons.

Interviewer: Thanks, that sounds really interesting. Do you feel that the variety of previous experiences helps you in this environment?

Respondent: Definitely. You learn to adapt, ask the right questions, and understand how different people think. And that, I think, is key for an analyst – to be a bit of a translator between worlds.

Interviewer: Great, so we can start the experiment we've prepared. Here's your assignment, and once you're ready, I'll start your respondent.

Participant-Respondent Interview (ChatGPT)

(https://chatgpt.com/share/67dae0a1-32a0-8008-b49e-48f7dd4e75d0)
Interviewer (Participant):
Thank you for taking the time to talk to me. I'd love to talk to you about your experience with the current approval system and what you expect from the new one. Can we start by telling me what typical tasks you perform on a daily basis in it?

Respondent:
Of course. In the current approval system, I process and approve various documents on a daily basis, especially budgets, purchase requests and internal financial guidelines. I monitor the status of individual requests to see if they have already gone through the entire approval process or if they are stuck somewhere. I also often deal with returning requests for rework when certain conditions are not met or important information is missing. In addition to approval and control, I also work with the system to prepare reports on how much expenditure was approved in a certain period or how long individual processes take. Within the team, we also communicate about specific requests, both with our department and with other approvers.

Interviewer (Participant):
That sounds like a fairly broad use of the system. What do you think are the main problems with the current solution? What most often slows down or complicates your work?

Respondent:
Yes, we use the system very intensively, but unfortunately it has several major problems that complicate our work rather than make it easier. The first major problem is its slowness and outdated interface. It often happens that the system responds slowly, requests take a long time to load, and sometimes the approval process gets stuck because the request is not displayed to the next approver in line. The second problem is the lack of automation - we have to

manually check whether all the documents are attached to the request, and it often happens that something is missing, which prolongs the process. The third major shortcoming is the lack of effective communication directly in the system. When we need to add information or explain certain items, we have to deal with it via email or other channels, which slows down decision-making and increases the error rate. We also have other problems, such as limited integration with other company systems, but we can get to that later.

Interviewer (Participant):
I understand, these are pretty major hurdles. Let's look at each of these issues separately. I'd start with the slowness and outdated interface - what specifically annoys you the most about it? Is it the design, the navigation, the amount of clicking, or something else?

Respondent:
What annoys us the most is a combination of several things. The design is really outdated and unintuitive, which means that it is difficult to navigate the system, especially for new users. Navigation is complicated - some functions are hidden in confusing menus, so for example, to simply approve a purchase request, we have to click through several screens. Another big problem is the amount of clicking. For every small action - such as opening a request, adding a comment or returning it for revision - we have to confirm multiple steps, which is tedious. And finally, there is the speed of the system itself, which is unsatisfactory. It often happens that requests take several seconds to load, in worse cases even longer, and when more people are moving through the system at once, performance deteriorates even more. Combined, this means that routine tasks that should take a minute take many times longer.

Interviewer (Participant):
That sounds like a really frustrating way to work. If you imagine the ideal user interface for a new system, what do you think it should look like? What would make your job the easiest?

Respondent:
The ideal user interface should be clear, modern and, above all, intuitive. It would help us the most if the new system had a dashboard where we could see all the important information at a glance - for example, how many requests are waiting for approval, which requests are delayed or where information needs to be added. Navigation should be simple, ideally with the ability to quickly search and filter requests by various criteria, for example, by approval status or request value. Another great feature would be the ability to perform approvals with one click directly from the main overview, without having to click deep into the system.
I would also appreciate a responsive design that would work well on mobile devices so we could approve requests on the go. If the system could display related information in one place – for example, when approving a purchase request, it would immediately show the available budget and previous approved purchases – it would save us a lot of time. Overall, the point is to make the system as efficient as possible, minimize unnecessary steps, and allow for fast work without frustration.

Interviewer (Participant):
That makes sense. You mentioned that you would appreciate one-click approval and better visibility into related information. What other specific improvements would make the approval process itself easier for you?

Respondent:

Another improvement that would make our work easier would be the ability to set up automatic approval rules. This means that certain types of requests that meet predefined criteria (e.g., an amount up to a certain amount or an approved supplier) could be approved automatically without the need for manual intervention. A bulk approval feature would also help, where we could mark multiple requests at once and approve them in one step, instead of having to open each one separately.
A big improvement would also be the ability to communicate directly within the application – for example, chat or comments on a specific request, so that ambiguities can be quickly clarified without having to switch to email or another tool. It would be ideal if the system could alert users to missing documents or discrepancies before sending them for further approval, to minimize the number of applications being returned.
Finally, it would be useful if the system had better integration options with our accounting and ERP system, so that budgets, contracts or previous orders could be directly seen without the need for manual searching. Overall, the point is to make the approval process as fast, as manual and as transparent as possible.

Interviewer (Participant):
These are very useful suggestions. When we talk about automating and validating requests, what specific conditions or rules should the new solution be able to automatically check to reduce the number of returned requests?

Respondent:
The new solution should be able to automatically check several key conditions that often lead to requests being returned and unnecessarily delaying the process. The first condition is checking for mandatory attachments – for example, for purchase requests, the system should automatically verify that all required documents, such as a supplier quote or approved budget, are attached. If something is missing, it should notify the user before the request is sent.
Another important check is budget compliance. The system should automatically check whether the request exceeds the available funds in the relevant budget category. If so, it could display a warning or directly offer an alternative approval path (such as escalation to senior management).
The system should also check the approval authority of the approvers – that is, whether the person can approve a specific type of request and amount. It often happens that requests are mistakenly sent to the wrong people, which delays the process. This also includes automatic routing of requests according to the approval matrix, so that each request goes straight to the right place without the need for manual settings.
Duplicate checking would also be useful, where the system could recognize whether a similar request has already been submitted, which would help eliminate errors and unnecessary expenses. And finally, it could verify the accuracy of billing data, for example by validating the supplier's ID number against external databases. This would reduce error rates and ensure that requests are complete and correct the first time they are submitted.

Interviewer (Participant):
This makes sense, automating these checks would certainly help streamline the process. If we look at the approval process itself. How many approval steps are typically needed and where do the biggest delays occur?

Respondent:
The number of approval steps depends on the type of request and its value, but generally ranges from two to five levels. A typical process begins with the requester submitting the

request, then it is reviewed by their direct manager, then goes to the finance department and, in some cases, to the company's management for approval. For larger amounts or strategic investments, the legal team or selection committee may also be involved in the process.

The biggest delays usually occur in two places. The first is when the request travels between individual approvers - it often happens that the document gets "stuck" with one person who forgets about it or is busy. Therefore, it would be great if the new system had automatic reminders or the ability to redirect the request to a representative in case the approver is absent.

The second critical point is the return phase for rework. If the application is not completed correctly or documents are missing, it has to be returned to the applicant, which significantly lengthens the entire process. As we have already talked about automatic checks, these could significantly reduce this problem. It would also help if the applicant received clear instructions on what to correct and had the opportunity to simply add information without having to submit a new application from scratch.

Overall, the new system should allow for better traceability of requests – for example, a visual display of who is currently handling a request and how long it has been there. This would help not only reduce delays but also improve transparency of the entire process.

Interviewer (Participant):
That's a good observation, traceability and automatic notifications could significantly speed up the process. How do you currently track the status of applications? Do you have any reports available or do you have to find the information manually?

Respondent:
Currently, tracking the status of requests is quite impractical. The system allows you to view a list of requests and their status, but it is not clear and we often have to search for information manually. There is no visual representation of where the request is currently located, so if I want to find out who it is stuck with, I have to open it and manually go through its history. This is not only time-consuming, but also inaccurate, because it is not always clear why the process has stopped.

Another problem is that there are no automatic notifications, so if a request is delayed, we have to contact the approver ourselves by email or phone, which is an unnecessary burden for everyone involved. We have the option to generate simple reports, but they are static and do not update in real time, so if I want the current status, I have to request it manually every time.

It would be ideal if the new system offered an interactive overview, such as a visual workflow where I could see at a glance who has a request on their desk and how long they have been holding it. Automatic reminders or an escalation mechanism that would alert management if a request is delayed beyond a set limit would also be great. This would make our work much easier and help speed up the entire approval process.

Interviewer (Participant):
I understand, so an interactive overview and automatic notifications would greatly help with efficiency. Speaking of reports, what specific information would you like to have available in the new system?

Respondent:
In the new system, we would need to have detailed and dynamic reports available that would allow us to effectively monitor and analyze approval processes. Key information we would like to see includes:

An overview of approved and rejected requests – ideally with the ability to filter by date, department, request type, and approver. We would like to quickly see how many requests were processed in a certain period and what the most common reasons for rejection were.
Approval process length – i.e. the average approval time for each type of request and identifying bottlenecks where the biggest delays occur. For example, we would like to see which approvers are experiencing the most delays and how long it takes them to respond.
Statistics on returned applications – how many applications were returned for rework, why, and what the most common errors were. This report would help us identify problem areas and potentially set up better automated checks.
Budget overviews – what approved expenses were agreed upon in a given period, how the budget is being used, and whether the limits of individual categories are not being exceeded. It would be ideal to connect to the accounting system so that budget data is updated in real time.
The effectiveness of individual approvers and departments – how quickly individual teams or individuals approve requests, who processes requests on time and who often blocks them. This report could be used to optimize processes and better distribute approval powers.
The most important thing is that these reports are interactive and dynamic, i.e. with the possibility of easy filtering, data export and visual display in graphs or dashboards. It would also be great if the system allowed to set up regular automatic generation and sending of reports by e-mail, so that the company management could be regularly informed about the status of approval processes without the need to manually search for data.

Interviewer (Participant):
This makes sense, clear and dynamic reports would definitely help with managing the approval process. You also mentioned integration with the accounting system. What other systems or tools should the new approval system be able to connect to make it as efficient as possible?

Respondent:
The new approval system should integrate with several key company systems to eliminate unnecessary manual data entry and enable smoother processes. The main systems it should be able to work with are:
Accounting and ERP system – This is absolutely essential for us. We need the new system to be able to retrieve current budget data and automatically check the availability of funds when approving purchase requisitions. It should also allow for easy integration with billing processes, for example so that approved requisitions can be automatically transferred to the accounting system as orders or invoices.
DMS (Document Management System) – Approvals often involve contracts, invoices and other documents. If the new system could connect stored documents directly from our DMS and ensure that approvers have the latest version without the need for manual uploads, it would make the job much easier.
Email and notification system – The system should be able to send automatic notifications via email, but ideally it should also support integration with communication platforms such as Microsoft Teams or Slack so that users receive notifications directly where they normally work.
Access Control System (Active Directory / SSO) – To facilitate user management, the new system should support Single Sign-On (SSO) so that users do not have to log in separately and can authenticate through the company's Active Directory or other central identity management system.

CRM system – If we approve orders or contracts regarding customers, it would be useful to connect to our CRM so that we can easily verify whether this is an existing client, what the current business conditions are with them, and whether there are no approved obligations for them.
BI tools (e.g. Power BI, Tableau) – For advanced analyses, the approval system should allow data export to analytical tools or have built-in support for connecting to BI platforms, where we could create more detailed reports and monitor trends in the approval process.
If the new system could integrate effectively with these tools, it would mean significant time savings, fewer errors, and smoother processes without the need to manually rewrite data between different systems.

Interviewer (Participant):
That's a good list of integrations that could make your work a lot more efficient. How would you like to manage user permissions in the new system?

Respondent:
User permission management should be as simple as possible, but flexible enough to allow for different levels of access based on user roles. Ideally, the new system should support the following permission management principles:
Connect to your company's Active Directory (AD) or other identity management system – Users should be able to log in using Single Sign-On (SSO), which would make account management easier and ensure that access is current within the organization. If someone leaves the company or changes positions, their permissions would be automatically updated.
Roles and permissions by function – Instead of manually setting up roles for each user, there should be predefined roles (e.g., requester, approver, financial controller, administrator) that determine what actions a given user can perform. For example, a requester could submit requests but not approve them, while the finance department would have access to all requests and budget information.
Approval Matrix and Dynamic Rules – Some permissions should be based on the amount or type of request. For example, a department head could approve requests up to a certain amount, but higher amounts would automatically escalate to higher management. This should be configurable so that the system automatically applies the rules without the need for manual review.
Ability to delegate permissions – If the approver is on vacation or out of the office, they should be able to easily delegate their permissions to a representative to avoid requests getting stuck. Ideally, the system could also set up automatic delegation rules based on user availability.
Audit logs and action history – To ensure transparency, the system should record who performed what action (e.g. approval, rejection, budget change) and when. This audit trail should be easily accessible to administrators and potentially exportable for internal controls or audits.
Overall, we want permission management to be as automated as possible and connected to the company structure, but at the same time flexible enough to allow specific settings according to the needs of individual users and approval processes.

Interviewer (Participant):
This makes sense, robust permission management is key to security and efficiency. How would you envision the onboarding process for new users?

Respondent:

The onboarding process for new users should be as simple and automated as possible, so that it does not require manual intervention from the IT department or system administrators. Ideally, we would imagine it like this:

Automatic role-based access assignment – If the system were connected to Active Directory (AD) or another HR system, it could automatically assign permissions to new employees based on their job title. For example, if someone joins as a financial analyst, the system would immediately grant them access to approve certain requests without the need for manual setup.

SSO and Single Sign-On – New users should be able to log in using Single Sign-On (SSO), so they don't have to create a new account or remember another password. Access would be managed centrally through the company's identity management.

Interactive onboarding in the system – Upon initial login, a new user could go through an interactive guide that shows them key features and explains how to work with the approval process. This could be in the form of short pop-up instructions or video tutorials.

Pre-configured dashboards and notifications – Every new user should see relevant information clearly from the start – for example, approvers would have an overview of requests awaiting their action on the main dashboard, while requesters would see the status of their submitted requests.

Easy way to contact support – If a new user has a problem, they should have a visibly available option in the system to contact support, whether via chat, FAQ section, or ticketing system.

Mentor roles or approval hierarchy – For more complex processes, the system could allow the assignment of a mentor or supervisor who would have access to the new user's approval activities and could help them in the beginning.

Audit and training for sensitive roles – Users with high levels of authority (such as a CFO or chief accountant) could undergo a short training or confirm familiarity with the system rules before activating their account.

Overall, onboarding should be as automated, intuitive, and manual as possible. The goal is for a new user to be able to start working in the system immediately after logging in, without having to wait for complex setup.

Interviewer (Participant):
That sounds like a very effective onboarding. You also mentioned user support. Would you prefer a separate help desk, direct contact with administrators, or a combination of both?

Respondent:
A combination of both would be ideal, i.e. both separate help and the possibility of direct contact with administrators, so that users can choose according to the situation. Specifically, we would appreciate:

Interactive help and FAQs right in the system – Users should have an easily accessible section with frequently asked questions and instructions, ideally contextually linked to individual functions. For example, if someone is filling out a form and doesn't know what a certain item means, they could click on the help icon and the relevant information would be displayed.

Quick Questions Search Bar – The ability to quickly find answers to common problems, similar to how full-text search works in knowledge databases.

Video tutorials and interactive guides – For new users, it would be great to have short instructional videos or a guide that walks them through the main features of the system step by step.

Ability to contact the administrator or IT support – If the user is at a loss or has a technical problem, they should be able to simply contact the administrator, ideally via an integrated chat or ticketing system, where they could describe their problem and monitor the status of its solution.
Community forum or internal discussion section – If the company has a larger number of users, it could be useful to have a place where employees can help each other by sharing experiences and advice.
Automatic diagnostics and troubleshooting of common problems – If the system detects that a user is failing to perform a certain action (e.g. repeatedly filling out a form incorrectly), it could offer automatic help or recommendations.
Overall, we would prefer a separate help desk for common questions and quick contact with administrators for more complex issues, so that the system is as self-contained as possible, but at the same time allows for a quick solution if necessary.

Interviewer (Participant):
This makes sense, the combination of separate help and direct contact options will cover a wide range of user needs. Do you need to maintain access to old requests, or should only a limited archive be available?

Respondent:
We certainly need to maintain access to old requests, but ideally it should be possible to set archiving rules so that the system does not become unnecessarily cluttered with old data. The most important archiving requirements would be:
Long-term access to historical requests – Some requests (such as investment purchases, long-term contracts or audits) need to be traced back several years. Ideally, we could set a retention policy based on the document type – e.g., routine purchase requests available for 2-3 years, but strategic investments for 5-10 years.
Archive searchability – Old requests should be easily accessible via search and filters to avoid having to manually sift through large amounts of data.
Automatic archiving of old requests – After a certain period of time, requests could be moved to a special archive where they would be available only on request, thus relieving the main database.
Export of old requests – We should be able to export historical data to a file (e.g. CSV, PDF) or connect the archive to an external DMS, where we could set our own retention rules.
Audit requirements – If we have a legislative obligation to retain certain documents for a legally required period, the system should allow for their secure storage and ensure that they are not accidentally deleted.
Overall, we would like to maintain access to old requests, but at the same time manage them efficiently to keep the system clear and efficient.

Interviewer (Participant):
This is a logical approach to make the archive useful but not burden the system. What are your security requirements for the new approval system?

Respondent:
We have quite strict requirements for the security of the new approval system, as we work with sensitive financial data and need to ensure compliance with legislative requirements. Key security requirements include:
Authentication and Access Control – The system should support Single Sign-On (SSO) and two-factor authentication (2FA) for critical roles, such as CFOs or high-threshold approvers.

Access should be managed based on roles so that everyone sees only what is relevant to their job.
Data encryption – All data should be encrypted, both in transit (TLS 1.2+) and at rest (AES-256) to minimize the risk of sensitive information being leaked.
Audit logs and activity tracking – Every action in the system (approval, rejection, data change) should be recorded in an audit log, which should be available to administrators and possibly exportable for internal or external audits.
Protection against unauthorized changes – The system should have protection against unauthorized manipulation of data, for example, the ability to digitally sign approved documents or prevent modifications to approved applications without additional authorization.
Security roles and the principle of least privilege – Each user should only have the permissions they absolutely need to minimize the risk of unauthorized access to sensitive data.
Protection against attacks (e.g. DDoS, SQL injection, XSS) – The system should have built-in protection mechanisms against common cyber threats, including regular security updates and vulnerability testing.
Regular backups and a disaster recovery plan – Data should be backed up regularly and there should be a plan for system recovery in the event of an outage or cyberattack. Ideally, we would want geographically separated backups with the ability to restore quickly.
Compliance with legislative requirements – The system should comply with GDPR (especially regarding the retention and protection of personal data) and other relevant regulations for financial data.
Security notifications and monitoring – Administrators should receive alerts about suspicious activity, such as repeated failed login attempts or unusual approval patterns (e.g., if someone suddenly approves a suspiciously large amount).
Overall, we want the system to be secure, but at the same time user-friendly, so that security measures do not complicate everyday work. We are looking for a balanced approach between data protection and the efficiency of approval processes.

Interviewer (Participant):
This is a sensible approach – to ensure a high level of security, but at the same time not to complicate everyday use. What should be your requirement for system availability?

Respondent:
We require high system availability because approval processes are crucial to the normal operation of the company and any outage can cause delays in important purchases, contracts or investments. Specific availability requirements include:
SLA of at least 99.5% – Ideally, we would like a guaranteed availability of at least 99.5%, which means a maximum of a few hours of downtime per month. If the system is cloud-based, it should have a robust infrastructure that minimizes the risk of downtime.
Redundancy and failover mechanisms – If one server or data center fails, the system should have automatic failover to a backup environment to ensure uninterrupted availability.
Disaster Recovery Plan – There should be a clearly defined disaster recovery plan to ensure the system is restored in the event of a major outage (e.g. cyber attack, infrastructure failure). We expect the system to be fully restored within a matter of hours, maximum within one business day.
Regular backups – Data should be backed up at least once a day and there should be a geographically separated backup repository to protect data even in the event of a main data center crash.

Offline mode or emergency approval – Ideally, there should be an emergency mode that allows for at least basic approval even in the event of a system outage, for example in the form of email approval or temporary local access to necessary requests.
Monitoring and rapid response to incidents – The system should have automated monitoring of availability and performance with the ability to immediately notify administrators in case of problems. If an outage occurs, we expect a rapid response from technical support and transparent communication about the progress of the solution.
Overall, we want the system to be reliable, robust, and resilient to outages, because any delay in the approval process can have a real impact on the business.

Interviewer (Participant):
This makes sense, high availability and rapid recovery are key to business continuity. Do you have a preference as to whether the new system should be cloud-based or on-premise? Or would you like a hybrid solution?

Respondent:
We prefer a cloud solution, but with an emphasis on security, availability and the possibility of integration with our systems. The main reasons for a cloud solution are scalability, lower demands on internal IT management and higher availability. Specifically, we expect:
Secure enterprise cloud – The system should run on a proven cloud platform (e.g. Microsoft Azure, AWS, Google Cloud) with security and compliance certifications (e.g. ISO 27001, SOC 2, GDPR).
High availability and SLA – We expect an availability guarantee of 99.5% or more, automatic backups and failover mechanisms to minimize downtime.
Automatic updates and maintenance – A cloud solution should provide regular security updates and new features without the need for intervention from an internal IT team.
Encryption and data protection – We require end-to-end encryption and ensuring that data is not stored in the wrong jurisdictions. Ideally, the ability to choose where the data is physically stored (e.g. in the EU due to GDPR).
Integration with on-premise systems – Even though we prefer the cloud, we need the system to be able to connect with our on-premise ERP, accounting system, or Active Directory.
Hybrid access option for sensitive data – If we were working with very sensitive information, we want to be able to store certain parts of the data locally (on-premise), while regular operations would take place in the cloud.
If we had to choose on-premise, we would have to provide our own infrastructure and management, which is expensive and less flexible. A hybrid model could be an alternative if strict security requirements had to be met, but the primary preference is a modern, secure and well-integrated cloud solution.

Interviewer (Participant):
This makes sense, a cloud solution with the option of a hybrid approach combines flexibility with security. Finally, if you had to summarize the three most important features that a new approval system should have, what would they be?

Respondent:
If I were to summarize the three most important features that a new approval system should have, they would be:
Efficiency and automation – The system should minimize manual steps and streamline the entire approval process. This means automatic checks for the correctness of requests,

intelligent routing to the right approvers, the possibility of mass approval, and a well-designed workflow that eliminates unnecessary delays.
Transparency and intuitive user interface – The dashboard should provide an immediate overview of requests, their status and any delays. Users should be able to navigate the system easily and complete approvals quickly and without unnecessary clicks. It is also important to be able to visually display the process and efficiently search through the request history.
Security and Reliability – The system must meet high security standards, including two-factor authentication (2FA), data encryption, audit logs, and strict role-based access control. We also expect high availability (99.5% or more), regular backups, and rapid recovery in the event of an outage.
If the new system can meet these three key requirements – it will be efficient, intuitive and secure, then it will be a successful implementation for us that will significantly improve the approval processes in the company.

Interviewer (Participant):
Perfect summary, I think we now have a very clear idea of your requirements and expectations. Thank you for your time and detailed answers. All this information will help in designing a new approval system to best suit your needs. If you have anything else important to think about, please let me know.

Respondent:
Glad it happened! I'm glad we were able to go through all the key requirements and that it will help to design an effective solution. If there are any other details or specific situations that we should consider, I will definitely be in touch. Thanks for the well-conducted interview and I look forward to the next steps in the process of implementing the new system!

Conclusion of the interview with participant 1

Interviewer: Thank you again for sharing the interview sample. It was very interesting to see how it actually works. I would like to follow up a bit and go into more depth about your requirements gathering methodology and strategy. I will start with a simple question – what development methodology do you currently use in your company?

Respondent: We work agilely, specifically according to the SCRUM methodology. We have classic sprints every two weeks, regular stand-ups, sprint planning and retrospectives. But I have to say that our approach is more of a pragmatic agile one – we adapt to the projects. Sometimes it's more of a "scrum-ban", sometimes we make compromises when the situation requires it. But we hold agile principles such as iterative development, constant feedback and teamwork quite firmly.

Interviewer: And how does the agile approach work for you in the role of analyst?

Respondent: Personally, I like it. Mainly because I'm not "done" after one phase. I can be with the developers and the business continuously, react to changes and adjust the specifications based on feedback. It's more about a dialogue than a one-off document, which makes sense to me.

Interviewer: Let's talk about your strategy. When you interview stakeholders, do you have a specific approach or tactic to get really high-quality requirements?

Respondent: Yes, I have a few principles that I follow. The first thing is that I never jump straight into "what you want." I try to understand the context first – what they do, what their workday looks like, where the frustrations arise. I ask a lot about specific cases, not general wishes. I often use questions like "Can you show me the last time this didn't work?" or "What would happen if this system disappeared completely today?" This helps uncover real needs.

Interviewer: That sounds like a good way to get below the surface. Do you have a procedure for structuring your interview questions?

Respondent: I usually start with an open-ended question to let them talk. Once I feel we are in an area where there are recurring problems, I start to structure it more. I use the 5x Why method, for example, to get to the root cause. I also create a map of actors and processes, and then validate it with the respondent during the interview. It is important not to be dogmatic – each person reacts differently, so sometimes I have to adjust my style and pace.

Interviewer: And when you set the goal of the interview – what types of requirements are you trying to get? Do you focus only on functional ones, or do you address other types as well?

Respondent: I get a combination of functional, non-functional and technical. Functional are the most common – what should the system do, what steps the user takes. But very often it happens that the "non-functional" requirements are the most critical ones – for example, performance, availability, security. If I don't ask about them specifically, they usually remain hidden. For example, when requesting fast approval, I have to check what "fast" means – two seconds? two minutes?

Interviewer: Do you also have a way to access technical requirements that the user does not know directly?

Respondent: Yes, this is where collaboration with developers helps. I usually "mediate" what the user says, and the developer translates it into a technical solution. But I also ask things like: "What other systems do you use? What does it need to communicate with?" or "If we had to solve this from scratch – what would help you the most?" Technical requirements often arise from constraints or from existing infrastructure. So it's about listening and connecting information from different sides.

Interviewer: When you prepare for an interview, do you also plan hypotheses or scenarios?

Respondent: Definitely. Before the interview, I usually prepare working hypotheses – for example, "I think the problem is that they don't have an overview of the applications". I then either confirm or refute these during the interview. Sometimes I also prepare model scenarios, for example, what the ideal process could look like, and then I give it to the respondent as a hint – for example: "If the system immediately pointed out the missing attachments, would that help you?"

Interviewer: Do you have a mechanism during the interview to validate understanding of the requirements?

Respondent: Yes, validation is key. I do it all the time – I paraphrase, I make sketches and point them out, I ask "Am I understanding correctly, right…?". After the interview, I also often write a summary and send it back for approval, or we share it in a workshop where we look at specific suggestions. The more feedback, the more accurate the output.

Interviewer: That makes sense. Is there anything you would describe as your strength in conducting interviews?

Respondent: I think I can put people at ease. I don't go into interviews as "someone from IT", but as a partner who wants to help them. I try to be human, ask questions openly and show that I am really interested in their problem. This often gains my trust and they open up more than if it were a purely formal interview.

Participant 2

Beginning of the interview with participant 2

Interviewer: Hello, thank you for taking the time to talk to me. Before we begin, I would like to ask – how are you today?

Respondent: Hello, I'm glad you took the time. I'm fine, thanks for asking. Today is quieter than usual, so it's a nice change.

Interviewer: I'm glad to hear that. So I'd like to start with the first question - how long have you been working as a business analyst?

Respondent: I've been working in business analysis for eight years now. It's a role that I still enjoy, although it has of course changed a lot in that time. Every project is different and that's what's interesting about it.

Interviewer: And how did you actually get this role?

Respondent: I started after school as an internal project coordinator. At that time, I worked a lot with analysts and gradually took over some of their tasks - for example, analyzing requirements or preparing use cases. After some time, a junior analyst position became available and I went full-on into it. It actually came quite naturally.

Interviewer: So you originally took on a different position?

Respondent: Yes, exactly. The original position was very project-oriented, but even then I felt that I enjoyed listening to people more and looking for ways to improve or change things. The analytical level was close to me even then.

Interviewer: And in what types of projects do you have the most experience? I mean IT projects, process analysis or strategy...

Respondent: I definitely have the most experience with IT projects. I spent a lot of time collecting and analyzing requirements for the development of internal systems – for example, CRM, approval processes or workflow management. But I also worked on purely process projects, for example when we mapped and optimized internal procedures after a department merger. Not so much with strategy, but even there I occasionally intervened as part of larger transformation initiatives.

Interviewer: What about your professional background – have you worked in different companies, or do you have longer-term experience in one organization?

Respondent: I worked for two companies. I was in the first one for about three and a half years, it was a medium-sized IT company. And for the last more than four years I have been in my current company, which is much larger – we have over 500 employees and many branches around the country. It is a different dynamic, but you learn different things here than in a company where everyone knows each other by name.

Interviewer: That sounds like a varied experience. So you currently work for a large company?

Respondent:
Yes, that's right. It's corporate, but I have to say we have a fairly agile approach, so it's not exactly the typical "heavy-handed" environment that people often imagine. It has its pros and cons, but so far I'm happy.

Interviewer: Thank you, that's great.We can start the experiment we've prepared. Here's your assignment, and once you're ready, I'll start your respondent.

Participant-Respondent Interview (ChatGPT)

(https://chatgpt.com/share/67e053b7-ed08-8008-ae97-c7ebbf1f4014)
Interviewer (Participant):
Hello, thank you for taking the time to talk to me. I would like to talk to you about the approval system that you currently use. Could I first ask – what types of requests or documents do you personally approve most often in it?

Respondent:
Hello, of course, I'd be happy to take the time - this is a pretty important change for us. Personally, I most often approve purchase requests, budget changes and contracts. For those purchase requests, it's typically about approving higher amounts above a certain limit, or requests that have an impact on cash flow. For budget changes, we mainly deal with transfers between budget items or requests for increases. And for contracts, it's about the financial aspect - whether they are in line with the budget, whether the payment terms are set correctly, or whether there are any hidden obligations that would cause us problems in the future.

Interviewer (Participant):
Thank you, that's super clear. And... just to clarify, when you're approving purchase requests, what does the process look like in the current system? I mean, like... what steps does it typically go through before it goes through you or before you approve it?

Respondent:
Sure, I'll describe it to you from the perspective of a typical purchase request. The process usually starts with the client in a department, such as marketing or IT, who enters the request into the system. They fill in the purpose, amount, and possibly attach offers or other documentation. Then the system automatically forwards it to the head of the department for initial approval. If the amount is higher or it involves an investment, it is forwarded to me for financial approval. I check whether it is in line with the budget, whether we have the funds for it, and if so, whether the billing code is set correctly. After my approval, it usually goes to the COO or CEO - depending on the amount and importance of the purchase. Only then is the

order issued. However, the whole process is quite inflexible - approval paths cannot be easily changed and the system does not provide a good overview of where things are "stuck".

Interviewer (Participant):
Oh, right… so the approval paths are fixed and can't be adjusted much to suit a specific case. May I ask: how often does it happen that you have to deal with something outside the system, for example by email or phone, because of this inflexibility?

Respondent:
Unfortunately, quite often, I would say that in more complex or unusual cases, in about 30-40% of cases, something is resolved outside the system. Typically when a request falls under multiple budget lines, or when the responsibility for approval changes - for example, due to a vacation or a personnel change. The system does not have the flexibility to change the approver, so people forward it by email or call, and then someone still has to manually enter it back into the system so that it has a formal record. This creates confusion, delays and, most importantly, traceability is lost - in some cases it is not clear who actually approved what and when, unless it was resolved directly in the system.

Interviewer (Participant):
Yeah, I get it... that must be pretty frustrating, especially when it comes to larger amounts or something important. And um... when you say traceability is a problem. What specifically are you missing from the current system? I mean... would you like approval history, timestamps, or something else?

Respondent:
Exactly, traceability is one of the things I really miss about the current system. Ideally, I would need to have a complete approval history – who approved it, when, with what comment, if there were any changes during the process, and most importantly, what status the request is in right now. Right now, the system shows who is supposed to approve it, but as soon as someone moves it forward or something changes, the overview disappears. Notifications are also missing – I often get a request late because someone didn't notice that they had to approve it. So, in addition to the history, automatic notifications and a dashboard where I could see all the requests waiting for me or someone else in the process would be very helpful.

Interviewer (Participant):
That sounds like something that would really make work easier, and when you talk about notifications, what method of notification would suit you best? I mean, by email, in the system, or maybe even some push notifications to your mobile phone, if the new system can do that?

Respondent:
Ideally, a combination. Email is still the basis, because we all have it open all day and we can quickly find it. But at the same time, I would like notifications directly in the system – something like a task overview on the main screen after logging in, where I can see what's waiting for me. And if the new system also had a mobile app, push notifications would make sense mainly for urgent requests – for example, when something is pending and just waiting for the final approval. But it should be possible to filter it somehow, so that we don't get overwhelmed with dozens of messages a day. The worst thing is when it's overlooked just because it got lost among other emails.

Interviewer (Participant):
Jojo, that makes sense, thank you. So there should be an overview, but also the possibility to somehow filter or adjust what you want to receive. And speaking of urgent requests, how do you even know that something is urgent? Is there a switch in the system for that or does it have to be handled externally?

Respondent:
Unfortunately, the current system is very weak in this regard. There is no switch or "urgent" flag, so if someone needs something approved quickly, they have to write to me on the side – usually by email or call. This of course leads to some things "jumping the queue" and others being delayed, even if they are no less important. It also sometimes happens that someone marks something as urgent, but from a financial point of view it does not seem so – we lack the ability to set priorities within the process. The new system should definitely allow some kind of urgency flag when entering a request, ideally with a short justification, and at the same time, as an approver, I would like to be able to adjust the priority if I see that it does not match reality.

Interviewer (Participant):
Oh, right, so it would actually be ideal if you could set the priority when creating the request, but at the same time, you, or maybe someone from finance, could reevaluate the priority? And maybe a stupid question - but do you think it would be better to have just a few priority levels like low, normal, high, or would you prefer some more precise definition?

Respondent:
No, that's not a stupid question at all, on the contrary - this is quite key. For me, three levels would be more than enough: low, normal and high. The point is to make it as simple as possible and for people to actually use the priority - when there are too many options, they either don't know how to deal with it or fill it in randomly. More important than the number of levels is to clearly define what each level means - for example, "high" is something that has an impact on an ongoing project or production and needs approval within 24 hours. And of course, the priority should be shown to everyone who enters the process so that they can act accordingly.

Interviewer (Participant):
Great, thanks. And moving a little further, you mentioned that sometimes a request concerns multiple budget items. Can I ask how you currently handle that in the system? Is there a way to enter multiple budget lines there, or do you have to work around it manually?

Respondent:
Yes, this is one of those common pain points. In the current system, it is technically possible to enter more budget lines, but it is quite clumsy - the interface is unclear, there is no validation, and it is also impossible to properly determine which amount goes to which code. It often happens that the client makes a mistake or simplifies it by putting it all under one line, even if it is not correct. We then have to manually correct it in accounting. In some cases, people prefer to create two separate requests, which is confusing and kills the overview of the total amount. So ideally, I would like the new system to allow you to simply add more budget items, divide the amount, have a clear summary, and also validation - for example, the total amount must correspond to the sum of the items.

Interviewer (Participant):
Sure, so I understand the possibility of multiple items, clearly, with a checksum. And when you talk about accounting, do you have any connection of the current system to the accounting or ERP system, or do you rewrite everything manually?

Respondent:
Unfortunately, the connection is very limited. The current approval system works more or less as a standalone application – it does not communicate directly with the ERP system, which is Helios in our case. This means that after the request is approved, our team has to manually rewrite the data into Helios: amount, budget lines, center, purpose and other data. It is time-consuming and prone to errors. Moreover, if something changes later – for example, the approved amount or budget allocation – it has to be manually edited on both sides. In an ideal world, the new system would be able to either export directly to Helios, or at least provide a structured output (e.g. XML or another standardized format) that we could upload. And the best thing would of course be an online connection via API, if Helios allows it.

Interviewer (Participant):
Oh, right, so now there's a lot of manual work and errors occur. And in connection with that, I wonder what about any changes after approval? Once the request is approved and then the amount or budget is adjusted, how do you handle that now? Do you have a process for that, or is it handled ad hoc?

Respondent:
Yes, that's another weak point of the current system. Officially, every change after approval should go through the approval wheel again - but the system doesn't fully support this. So, for example, when the amount is adjusted, we usually deal with it ad hoc: the client notifies us by email, we note it somewhere and then manually correct it in accounting. But the original record remains in the system and the change is not recorded anywhere. There is no revision, no audit trail, and sometimes it is not even certain who actually approved the change. Ideally, I would like the new system to have the option to return the request for approval if something significant changes after approval - the amount, budget allocation, or perhaps the supplier. And at the same time, a record of all changes, who made them, when and why.

Interviewer (Participant):
Sure, I understand, so the ability to go back to approval and have a history of changes. That makes a lot of sense. And can I also ask how often do such changes actually happen after approval? Is it more of an exception, or is it something you deal with regularly?

Respondent:
I would say that this is not entirely exceptional – it happens in about 15 to 20% of requests. Sometimes it is a small thing, such as rounding off the amount after receiving the final offer from the supplier, but it often happens that the total amount, supplier or budget coverage changes. For example, if the delivery is delayed and it is moved to another month or quarter, we have to move it under a different budget framework. Sometimes it also turns out that the original assignment was not entirely accurate and the purchase needs to be divided between two different budget items. So yes, it is something that we deal with regularly, and the absence of a proper mechanism in the system costs us time and the certainty that we have it correctly recorded.

Interviewer (Participant):

Yeah, I understand, if it's this frequent, it would definitely be worth having it directly in the system. And speaking of those frequent situations, I wonder if you have recurring types of requests? Something that is approved regularly with the same parameters? And if so, is it always solved from scratch, or do you copy or template it somehow?

Respondent:
Yes, we have recurring requests – for example, regular service orders, rent, license fees, or various service contracts. They are usually for the same suppliers, similar amounts, and the same purpose. Unfortunately, in the current system, we don't have the option to create templates or copy a previous request, so people enter it manually from scratch every time. Often, someone opens an old request aside, copies the texts into Word or Notes, and then rewrites it back, which is not only tedious, but also prone to errors – for example, if they forget to adjust the date or amount. In the new system, I would definitely like the option to create a template or duplicate a previous request with the option to edit before sending. That would save us a lot of time.

Interviewer (Participant):
Oh, right, so the ability to save as a template or duplicate a request would be very useful. And if there were templates, do you think they should be more personal to the person who created them or shared within the department?

Respondent:
Ideally, both. It would be nice for a regular user to have their own personal templates – for example, when they repeatedly enter the same type of purchase or payment. But at the same time, as a manager, I would appreciate the possibility of having shared templates at the department or company level, especially for things that should have a uniform structure. For example, a template for a license fee or car rental – so that everyone enters it the same way and someone doesn't have to fix it every time. Of course, it should have some permissions – who can create templates, who can change them, so that there is no chaos.

Interviewer (Participant):
Yeah, that makes sense, both personal and shared, but with adjustable rights. And speaking of rights, what about approval permissions? Are they somehow fixed according to functions, or can someone adjust them, for example when there is a crowd?

Respondent:
Unfortunately, permissions are now quite fixed – approvers are set according to the organizational structure and changing them is not easy. When someone is on vacation or sick, it has to be handled manually via IT or a system administrator, who can temporarily reset the approval. Sometimes we don't even have time and the request remains "hanging" until the person returns. In practice, we still resolve it by email or phone, and as I said before, there is zero traceability in this. I would definitely welcome the option of delegation in the new system – ideally so that the approver could set up a representative for a certain period, and so that it would be clear that the representative is approving, not the original person. It would also be nice to be able to approve from a mobile phone or email when someone is out of the office, but of course, taking security into account.

Interviewer (Participant):
Yeah, that sounds like a really important thing, the ability to be present outside the office. And since you mentioned security, I just want to make sure how important security is to you

for approvals? Do you have requirements for two-factor authentication or an audit trail for logins?

Respondent:
Security is key for us, especially with financial approvals, where hundreds of thousands or even millions are often involved. At a minimum, I would expect the new system to support two-factor authentication when logging in – ideally via a mobile app or an authentication app like Microsoft Authenticator. For more sensitive actions, such as final approval over a certain amount, it would make sense to require confirmation during the approval itself, not just during login. And there must certainly be an audit trail – who logged in when, from where, what they edited or approved. Not only for internal supervision, but also for possible external control – for example from audit or compliance. We really can't do without it in accounting and finance today.

Interviewer (Participant):
I understand, thanks, so two-factor, confirmation for larger amounts and detailed audit logs. And speaking of amounts, do you have any limits in the current system, when, who can approve what? And are those limits set by function, amount, or perhaps type of expense?

Respondent:
Yes, we have limits, but they are quite rigid in the current system. They are set according to function and amount – for example, a department head can approve requests up to CZK 100,000, I as a financial manager can approve requests up to CZK 500,000, and above that, approval goes to the operations or general director. The problem is that these limits are not linked to the type of expense or budget framework – so even a small investment outside the budget can go through a lower level, which is not ideal. Moreover, when someone exceeds their limit, the system often does not recognize it automatically, and the approval process gets stuck or has to start all over again. Ideally, the new system should be able to set limits in combination – according to role, amount, type of expense and perhaps even center. And most importantly, it should be easy to manage, not just through IT.

Interviewer (Participant):
I understand that, so a combination of multiple conditions and ideally automatic control of limits in real time. And just to give you an idea, how many approval levels do you typically have? I mean from the assignment to the final approval, how many people go through it on average?

Respondent:
On average, there are three to four levels. The first is always the client, then it goes to his direct superior – the head of the department. If it is a larger amount or the cost exceeds the budget, it is sent to me for financial assessment. And above a certain limit, it must then be confirmed by the operations or general manager. For investments or contracts with an impact on several departments, it can be five or six steps, when additional approvers are added. The problem is that these levels cannot be managed dynamically – for example, when approving a project purchase that is to be divided between two centers in terms of budget, the second manager should also see it, but the system cannot do that. In the new system, I would definitely like the route to be compiled automatically according to the parameters of the request – type of cost, center, amount, etc.

Interviewer (Participant):

Yeah, right, so dynamically building an approval path based on rules would help you a lot. And when something is being approved, what type of information do you need for approval to make a decision? Like what do you check before you click it off?

Respondent:
I mainly check three things. The first is the budget side – whether we have coverage in the budget for it, whether the request is correctly included under the budget item, and whether we will not exceed the center limit or the overall framework. The second thing is the factual justification – whether it makes sense for the expense to be incurred, whether the relevant documentation is attached to it (e.g. offer, contract, price calculation), and that it is not an obviously unnecessary or poorly timed expense. And the third area is formal – whether the fields are filled in correctly, the center, accounting dimensions, purpose, and possibly internal project or order numbers. For larger amounts, I also often compare whether there are no duplicates – for example, the same supplier and a similar amount are no longer in the system somewhere. So I need to have clear information in the system, ideally in one place, and the ability to quickly view attachments and related data without having to click through five screens.

Interviewer (Participant):
I understand, so having that information readily available and together is key. And regarding attachments, what types of documents do you attach most often? And do you have any restrictions on file format or size now?

Respondent:
We most often attach offers from suppliers, draft contracts, budget calculations in Excel, or price comparisons or email communication to clarify the assignment. The formats are mainly PDF, DOCX, XLSX and sometimes images, for example for equipment or graphic designs. As for restrictions - yes, the current system has quite strict rules. Sometimes it does not allow a certain type of file (e.g. ZIP or older formats), and we often encounter a size limit - I think around 10 MB per attachment. When we have larger contracts or multiple documents, we either have to reduce it or send it separately by email, which unnecessarily dilutes the information. Ideally, the new system should allow for a larger volume of data, support common formats without unnecessary restrictions and ideally also preview the attachment directly in the system so that you don't have to download everything just to see what's in it.

Interviewer (Participant):
Jojo, I totally get it, the preview option would definitely save time. And when adding more attachments, would it help if they could be categorized or labeled somehow? Like you could see right away what is a contract, what is an offer, or is it an unnecessary detail?

Respondent:
On the contrary, it would help me a lot. When there are multiple attachments to a request, it can sometimes be quite confusing to know what's what - especially when people don't care about file names. The ability to label or classify attachments by type - for example, as "contract", "bid", "budget", "communication", or "internal document" - would make work much clearer. Ideally, it would be possible to choose the type from a selection when uploading a file and then see it in the request overview. That way, I wouldn't have to open each file just to find out what's in it. And it would definitely be useful to be able to search by attachments - for example, to find all requests where a contract with a specific supplier is attached.

Interviewer (Participant):
That's great, thank you. So categorization and ideally also search by attachments. And just to go back to what you said earlier, when you approve a request, do you ever need to communicate with the client directly in the system? And if so, how do you handle that now?

Respondent:
Yes, quite often I need to clarify or explain some information – for example why a request is posted to a certain item, why there is no attached quote, or whether something has not been ordered before. However, the current system does not have a function for comments or messages directly to the request, so I have to deal with it externally – most often by email or phone. And this again loses overview and traceability, because the communication remains outside the system. Ideally, I would like the new system to allow writing comments directly to the request, so that it can be traced back, and for the person to whom the message is intended to receive a notification. And of course, for the comment to be visible to see who wrote it and when, so that it can be used as part of the approval history.

Interviewer (Participant):
Yeah, that makes sense, so basically an internal discussion on the request, with history, notifications and ideally an overview of who wrote what. And then I think, does it sometimes happen that more than one person needs to comment on a request without being an official approver? For example, an advisory vote or a comment from someone from another department?

Respondent:
Yes, this is exactly what happens quite often. Typically when it comes to a request that affects multiple departments – for example, an IT purchase that has an impact on security, or a marketing activity where lawyers are also involved. These people are not official approvers, but we need to know their opinion before it moves forward. However, there is no function for this in the current system, so we either deal with it on the side via email, or the whole thing is “officially” redirected through them, which unnecessarily drags out and complicates the approval process. Ideally, the new system should allow the involvement of so-called consultants or observers – i.e. someone who can add a comment or recommendation to the request, but does not have approval authority. And ideally, it would be visible in the request history who was approached and what they wrote about it.

Interviewer (Participant):
Okay, so having the ability to involve consultants without having to change the entire approval flow sounds very practical. Now for a slightly different question. Can you think of anything that the new system definitely shouldn't do? Something that really annoys you about the old one and you don't want it to happen again?

Respondent:
Certainly – I have a few things to say about this, but I'll start with the biggest one: the new system should definitely not be a "black box". The old one has the characteristic that once you submit a request, you have almost no chance of finding out exactly where it is, who is "sitting" on it, or why it got stuck. This is extremely frustrating, because when someone then urged it, we had to find out manually or through IT. So the main thing I don't want to see again is the opacity of the process. The new system should have a clear overview – who currently has the request on the table, what state it is in when it was last pushed, and ideally

also a prediction of who is next in line. And the second thing is inflexibility – I don't want a system where every minor intervention in the approval path or authorization has to go through a system administrator or even a developer. We need to be able to manage basic things ourselves, at least within our department.

Interviewer (Participant):
I completely understand that, a black box is really not ideal. So clear visibility of the status of the request, who's next, and ideally the possibility of some self-management without dependence on IT. And what about reporting? Do you currently use the outputs from the system for any analyses or overviews? And if so, are you satisfied with it?

Respondent:
Yes, we need outputs regularly – we make monthly reports on the number and volume of requests, how many of them were approved, how many were rejected, and how much is running across departments or budget categories. But the current system is not entirely satisfactory in this regard – it only offers a few preset reports, which we still have to export to Excel and process manually there. There is no sensible filtering by period, type of expense or status of the request. So ideally, I would like to have flexible reporting in the new system – the ability to choose the report parameters myself, export it to Excel, and even a simple visualization directly in the system. For example, graphs by the volume of requests per month, by department or category. And it would be great if the report could be scheduled – for example, to automatically receive it by email once a month.

Interviewer (Participant):
That sounds like a big time saver, having the flexibility to choose report parameters and receive them automatically. And the last thing that came to mind is, what about integration with other systems? We talked about Helios, but do you have any other systems that the new approval system should be able to connect to? For example, with ERP, accounting or some projects?

Respondent:
Yes, in addition to Helios, which is our main ERP and accounting system, we would also welcome connections to other internal tools. For example, our attendance and personnel system (which runs separately), because some requirements are related to personnel changes or benefits - it would make sense to have a unified database there. A connection to our document archive would also be useful - we use SharePoint, where we store signed contracts and other key documents. If the approval system could directly export or link to a specific folder on SharePoint, it would make archiving easier. And if the system could somehow communicate with the project tool, for example via project numbers or workflow, it would also be beneficial - especially for investment expenses, where we need to monitor budget spending by individual projects. So the more the system can integrate or transfer data with others, the better - especially if this eliminates manual rewriting.

Interviewer (Participant):
That's a great overview, integration with Helios, SharePoint, and the project tool would definitely save a lot of time and eliminate errors during transcription. And if I could summarize it now. The new system should have a clear approval process, flexibility in setting approval paths, attachment management and the ability to communicate directly in the system, delegation options, integration with internal tools, and easy reporting that could be customized. Is that all correct, or can you think of anything else that we should include?

Respondent:
That summary is absolutely accurate and covers most of what is bothering us in practice now and what we would need from a new system. I might add one more thing, which is rather soft, but in my opinion important – usability. The old system is cumbersome, illogical and complex for new users. When someone new comes to the company, it takes them a long time to understand how to navigate it. So I would really like the new system to have an intuitive interface, clear steps, help directly in the system and perhaps even the ability to see the requirements of others (at least within the department), if they have permission to do so. The fewer barriers to operation, the more people will actually use the system correctly – and that saves us all work.

Interviewer (Participant):
Great add-on, thanks. That's really important, an intuitive interface and ease of use so that everyone can get used to it quickly, especially when new people join. So besides all those technical requirements, user-friendliness should be at the top of the list. Thank you very much for all the information, it helped me a lot and I believe that based on that we can define the exact requirements for the new system.

Respondent:
It was a pleasure, thanks to you too – it was great that we could discuss it in such detail. I will be happy if the result is a system that will make sense not only for us in finance, but also for other users. And if you need to clarify something, add something or test a proposed solution in practice, feel free to get in touch. I keep my fingers crossed that you manage to put it together as well as possible!

Conclusion of the interview with participant 2

Interviewer: Thank you very much for the sample interview, it was very useful. Now I would like to move on to a few questions about your approach and strategy in gathering requirements. May I?

Respondent: Sure, I'm ready, ask questions.

Interviewer: What development methodology do you currently use in your company and how does it affect the way you collect requirements?

Respondent: We work within the framework of agile methodology, specifically in a combination of Scrum and Kanban. We have classic sprints, but at the same time we monitor some types of work continuously. The agile approach gives us flexibility, but at the same time requires us to collect requirements iteratively – not just once at the beginning of the project. I try to collect requirements so that they are “just enough” – that is, specific enough to be implemented, but not so rigid that they would make it impossible to change during development.

Interviewer: And what is your strategy when conducting a conversation with clients? How do you try to get the right requirements?

Respondent: First and foremost, I try to create a safe and open environment where the client doesn’t feel pressured. I start with general questions to understand the context – what their role is, what bothers them in the current system. Then I move on to more specific questions, often using scenarios or hypothetical situations – “Imagine that...” Active listening is also important – it’s often not just about what they say, but what they imply between the lines. I ask a lot of “why” – when someone wants a certain functionality, I want to know what’s behind it. This often reveals deeper needs to me.

Interviewer: Do you use any specific question structures, such as open vs. closed, control questions, or provocative hypotheses?

Respondent: Yes, I usually build the interview in layers. I start with open-ended questions to let the respondent describe the process in their own words. Then I move on to semi-closed questions like: “Can you describe how the process works step by step?” And finally I ask specific control questions – like “What happens when the approver is on vacation?” or “How would you know if something is missing in the system?” I also find so-called design questions to be very useful – “If you could design the system yourself, what would you definitely not want to miss in it?”

Interviewer: Do you focus on certain types of requirements during the interview?

Respondent: Yes, I try to map functional, non-functional and technical requirements in a targeted manner. Functional requirements usually come naturally – people talk about what they want to do. I have to be more active with non-functional requirements – I ask about system speed, reliability, notifications, mobile accessibility, etc. I usually only tune technical requirements after a conversation with developers, but I collect information that influences them – for example, integration with other systems, requirements for output formats or an audit trail.

Interviewer: In the three-way interview, it was clear that you often paraphrased the respondent's answers. Is this a common part of your strategy?

Respondent: Absolutely. I use paraphrasing to verify that I have understood what the respondent is saying correctly, and to show that I am really listening to them. Moreover, it often happens that when they hear their description in my words, they realize that they would have phrased something differently – and that leads to a more precise request.

Interviewer: How do you know when you have enough information? When do you know that a request is “done”?

Respondent: When I can formulate a requirement as a “user story” or use case, where there is a clear goal, actor, action and expected result – and at the same time I can imagine how the developer will program it, the tester will test it and the user will use it. This usually means that I have gone through the requirement from multiple angles: what should happen, when should it happen, what if it fails, what are the exceptions. When the answers start to get repetitive or vague, I know we are done.

Interviewer: Thank you very much. Is there any advice you would give to analysts who are conducting such interviews for the first time?

Respondent: Don't be afraid of silence. Sometimes it's enough to be silent for a while and the respondent will fill in the most important thing himself. And also – don't approach the interview as an "interrogation", but as a dialogue. The best insights often come outside the main questions – between the lines, in facial expressions or in what people take for granted. These are the most valuable fragments.

Participant 3

Beginning of the interview with participant 3

Interviewer: Hello, thank you for taking the time to talk to me. How are you today?

Respondent: Hello, thank you for the invitation too. I'm fine, a bit busy with work, but nothing
terrible. And you?

Interviewer: Okay, thanks. Just to warm you up – how long have you been working as a business analyst?

Respondent: It will be a little over two years now. I started right after school, so I'm still gaining experience.

Interviewer: And how did you actually get this role?

Respondent: Honestly, a bit by accident. Originally, I wanted to go into project management, but during an internship I got involved in requirements analysis for a smaller internal system and I was really hooked. Then I specifically went after analytical positions.

Interviewer: Interesting! And did you have any other job positions before that?

Respondent: Not really, just a few internships during school, mostly in smaller companies where a lot of tasks overlapped. But I started full-time as a business analyst right after school.

Interviewer: What types of projects do you have the most experience in? More likely IT, process analysis or strategy?

Respondent: Mostly it was IT projects – development of internal applications, customer portals or mobile versions. But sometimes it was also intertwined with process analysis, especially when we mapped current procedures in the company before designing a new solution.

Interviewer: And have you worked in different companies, or do you have long-term experience in one organization?

Respondent: I've already worked in three companies. They were always smaller startups, usually around 10-20 people. So I had the chance to see a lot of things up close and I was given responsibility pretty early on.

Interviewer: So you are currently also in a smaller company?

Respondent: Yes, it's still a small team. There are about fifteen of us now. I like that it's very dynamic and that I can touch on things outside of pure analysis.

Interviewer: Great, so we can start the experiment we've prepared. Here's your assignment, and once you're ready, we can begin.

Participant-Respondent Interview (ChatGPT)

(https://chatgpt.com/share/67e1c1ad-ef0c-8008-8b45-d9e734ef815a)
Interviewer (Participant):
Hello, first of all, I would like to find out what are the main functions of the current approval system that you use in your finance department?

Respondent:
Hello, the current approval system in our finance department is primarily used for electronic document circulation, mainly for internal approval of budgets, purchase requests, invoices and contracts. Basic functions include the ability to enter a request, assign an approval route according to the document type, notify approvers and subsequently approve or reject with a comment. It also allows limited monitoring of the approval status and export of approved documents for further processing. Although these functions cover basic needs, they have a number of limitations, which we can discuss in the next step.

Interviewer (Participant):
Thank you for the detailed description. Can you please specify what are the main limitations of the current system that you would like to see addressed in the new solution?

Respondent:
Of course, there are more limitations, but the most important one, in my opinion, is the lack of flexibility and user-friendliness. For example, approval routes are fixed and cannot be easily changed without IT intervention - which is impractical when the organizational structure or responsibilities change. Another big problem is the very limited connection to other systems - the system "lives by itself", so we have to manually rewrite data, for example, into an accounting or ERP system, which is time-consuming and increases the risk of errors. There is also a lack of any analytics or reporting - we cannot clearly see how many requests are in the process, how long they are being held where, or who is the bottleneck. Last but not least, the interface is outdated, unclear and practically unusable on mobile devices, which complicates the work especially for managers who often approve on the go.

Interviewer (Participant):
Thank you for your detailed explanation of the problems with the current system. Let's start with the first point, which is the flexibility of approval paths. What specific features would you like to have in the new system to manage approval paths more efficiently without the intervention of the IT department?

Respondent:
Ideally, I would like the new system to allow the user to set up and edit approval paths directly, for example via an intuitive graphical interface. This means that we in the finance department could define who has to approve what based on the type of request, amount, center or other parameters - without the need for programming. It would be a great advantage if the system supported conditional branching - for example: if the amount exceeds a certain limit, another level of approval is added. It should also be easy to change approvers or delegate authority, for example during vacation. The key is that these changes can be made quickly and without the risk of disrupting the entire process.

Interviewer (Participant):
Thank you for the precise description of the required functionality for managing approval routes. Now let's move on to the next point, which is integration with other systems. What specific systems would you like to integrate with the new approval system? And what types of data should be shared between these systems?

Respondent:
I would definitely like the new approval system to be fully integrated with our accounting system (we currently use Helios), ERP system and document management system (DMS). Ideally, the integration should work in both directions - for example, when a purchase requisition is created in the approval system, a corresponding record is automatically created in ERP and, after approval, it can be transferred directly to accounting. Conversely, invoices received in the accounting system should automatically appear in the approval system with the option of assigning them to an order and starting the approval process. The shared data would therefore include information about the requisitions (type, amount, center, purpose), approval status, order codes, billing data, suppliers and approvers. Furthermore, a connection with Active Directory would be suitable for managing access rights and automatically loading the organizational structure.

Interviewer (Participant):
Very good, thanks for specifying the necessary integrations and shared data. This will help us ensure that the new system will communicate effectively with other systems in your infrastructure.
Now let's focus on the topic of analytics and reporting. What specific reports or analytical functions would you like to have in the new system to better support decision-making processes in your finance department?

Respondent:
From a financial management perspective, I would definitely appreciate a set of predefined reports in the new system, as well as the ability to edit or create my own. Specifically, I would like to have an overview of all active and historical approval processes – who approves what, what status it is in, how long it takes at each level. I would also like to be able to monitor the total volume of approved requests by center, type of expense, supplier or project. A user-friendly dashboard with filtering options would be very helpful in regular financial reporting and evaluating budget performance. It would also be useful to have notifications about exceptions – for example, requests that exceeded the approved budget or that were “stuck” in the approval process longer than usual. Ideally, the system would also allow data export to Excel or connection to a BI tool that we already use in the company.

Interviewer (Participant):
Thank you for the detailed description of the required analytical functions and reports. This functionality will probably significantly increase the transparency and efficiency of the approval processes in your organization.
Now let's look at the user interface. What key features should the user interface of the new approval system have to make it as user-friendly and efficient as possible for your needs, especially for those who work frequently on the go?

Respondent:
The user interface should be simple, clear and responsive – so that it can be used comfortably on mobile devices or tablets. For users on the go, it is crucial that they can quickly view the

supporting documents for the request (for example, attached invoices or contracts) and easily approve or return it with comments. The entire process should take no more than a few clicks, without the need for complicated logins or clicking through several levels of menus. It would be very useful if the interface displayed an overview of all pending requests on the start screen, possibly with the option to filter by type or priority. I would also appreciate it if the approval process was visually represented – for example, in the form of a timeline or step-by-step guide, where you can see where the request is currently located and whose turn it is. We simply want an interface that does not delay, but helps.

Interviewer (Participant):
Thank you for your thoughts on the ideal user interface. It seems like you place a lot of emphasis on simplicity and efficiency, which is key to ensuring a smooth user experience in the field.
Now I would like to ask about data security in the new system. What security features do you consider necessary to be included in the new approval system, especially considering the sensitivity of the data that will be processed in it?

Respondent:
Yes, security is absolutely essential for us, because the system will process financial documents, contracts and personal data of employees and suppliers. First of all, I expect support for two-factor authentication, especially for approvers with higher powers. Furthermore, the system must be able to precisely manage access rights – i.e. who can see, change or approve what – ideally with a connection to Active Directory or another identity management system. All actions in the system should be auditable – i.e. recorded with a timestamp, user and type of action. Data encryption is also important, both during transmission and in storage. And of course, compliance with GDPR – the system must enable the management of personal data, its anonymization or deletion according to requirements. In addition, I would expect regular data backups, the possibility of recovery and, ideally, support for operation in the cloud with security standard certification (e.g. ISO 27001).

Interviewer (Participant):
A very thorough and comprehensive look at security requirements, thank you. This specification clearly defines the priority security areas that must be included to ensure the system complies with internal and external regulatory standards while protecting sensitive information.
Now I would like to focus on implementation and user training. What are your expectations regarding the implementation process of the new system and what training and support would you like to provide to users to best prepare them for the transition to the new system?

Respondent:
I expect the implementation to take place in phases and be accompanied by a thorough analysis of existing processes so that they can be optimized before the actual deployment of the system. The key for me is to create a test sandbox where we can first "touch" the new environment and test various scenarios. As for training, I would like it to be divided by role - i.e. different content for regular users, different for approvers and different for administrators. It would be ideal to combine face-to-face training for key users with online video manuals or interactive tutorials that will remain available later. Post-launch support is also important for us - I expect an available helpdesk at least in the first few months, ideally with the possibility of escalation directly to the system supplier. And to make the transition smooth, I would also

welcome the parallel running of both systems for a certain period of time, so that we can be sure that everything works as it should.

Interviewer (Participant):
Thank you, that's a very thoughtful approach. Phased implementation, sandbox testing, scaled training, and strong post-launch support are key factors for a successful transition.
I would now like to touch on the processes that go through the approval system. Can you please describe to me what the typical life cycle of a purchase request, for example, is like for you – from creation to final approval?

Respondent:
Of course, in our case, the typical lifecycle of a purchase request begins when an employee or center manager enters the request into the system – they fill in basic information such as the item description, estimated price, center, purpose of purchase and, if applicable, suggest a supplier. After sending, the request is automatically placed in a predefined approval path – first it is checked by the requester's superior, then by the responsible budget officer from the finance department, and if the amount is higher, it continues to management approval (typically the CFO or CEO). Each approver can approve the request, return it with comments or reject it. After final approval, the requester receives a notification and can place the order, with the approved request serving as the basis for the order in the ERP system. The entire process runs in the system, but due to its limitations, we often accompany the communication by email or telephone, which we would like to eliminate with the new solution.

Interviewer (Participant):
Thank you for the detailed description of the process. It is clear that the key is to improve communication and the approval process to make it as automated and transparent as possible. If we were to talk about possible improvements, what specific changes or features would you like to see in the new system to eliminate these difficulties and make the entire process more efficient and less dependent on manual communication?

Respondent:
I would definitely like the new system to automatically inform all involved parties about changes in the status of a request – ideally not only by email, but also via push notifications in the mobile application. The ability to add comments directly within the request is key for me, so that all communication remains in one place and is traceable. The ability to attach and share attachments (quotes, specifications, etc.) with the ability to version documents would also be useful. The process should be as automated as possible – for example, if the approver does not take any action within a certain period of time, the system will automatically notify them or redirect the request to their representative. Finally, I would welcome the ability to visually track the entire approval workflow in real time – ideally with the ability to click on each step and find out who is responsible and how long the request has been there. This would significantly reduce the need for manual investigation and tracing via emails.

Interviewer (Participant):
This makes perfect sense - such centralized and automated communication would significantly streamline the entire process and eliminate current inefficiencies.
Now I would like to ask about exceptions and non-standard situations. Are there cases in your department where the approval process needs to be bypassed, accelerated, or adapted in some way? If so, what types of situations are these typically and how would you like the new system to handle them?

Respondent:
Yes, such situations do occur from time to time – typically, these are urgent requests, for example, when a spare part needs to be ordered quickly due to a failure that would otherwise jeopardize operations, or when it is a purchase with a deadline of "the next day". In such cases, today we either call the approver or deal with it in parallel by e-mail, which means that the final documentation in the system does not correspond to the actual progress. In the new system, I would like it to be possible to mark the request as "urgent" – with a clearly defined accelerated approval scenario, for example via a shortened route or with automatic escalation. At the same time, the system should store information that it was an exception and allow for its later evaluation within the reporting. The process should also be adaptable – for example, the ability to manually change the approver or insert an extraordinary step, but only with the appropriate authorization and audit trail. Flexibility in these situations is key for us, but it must also be well controllable.

Interviewer (Participant):
Thank you for describing your needs for managing exceptional situations. Flexibility while maintaining controls and transparency is obviously very important to ensure smooth and efficient operations without compromising documentation or security procedures.
To conclude this segment of our conversation, I would like to ask what other key features or functionalities you would like the new approval system to include that we haven't discussed yet?

Respondent:
Good question – one thing that I still think is important and we haven't mentioned it yet is the ability to version and revise documents directly in the system. It often happens that a request is accompanied by a draft contract or technical documentation that is modified several times during the approval process, and we then trace back which version was actually approved. The new system should allow uploading new versions of files with the ability to compare or mark revisions so that it is always clear what exactly was approved. I would also like to see a connection to calendars – for example, so that approvers can automatically delegate their role during vacation. And the last thing – access to the system should also be possible offline with the ability to temporarily work in "on the go" mode and synchronize after connecting to the network. This would be especially appreciated by our managers who often work in the field outside the office or coverage.

Interviewer (Participant):
These are very important additional points - document versioning, connection to calendars, and offline mode support will definitely contribute to the professionalism and robustness of the system, and they also take into account the everyday reality of your work.
Now let's take a closer look at the user roles themselves. What types of users do you expect to have in the system, and what types of permissions or functionality should each role have?

Respondent:
In our environment, I would divide users into four main roles, each with clearly defined powers and functionality. The first and broadest group are requesters – employees who create requests. They need a simple interface for entering a request, attaching attachments, monitoring the status and providing feedback when the request is returned or rejected. The second group is approvers – executives or managers with responsibility for budgets. They must have a clear dashboard with assigned requests, the ability to approve, return with

comments, forward or delegate. The third group is financial administrators, which includes my department – they should have access to all requests within the company, the ability to set approval rules, monitor budget limits and prepare reports. The last group is system administrators, who are in charge of the technical administration of the system – user rights management, integration management, security and overall configuration. Each of these roles should have access only to the functions they really need, ideally adjustable according to the organizational structure.

Interviewer (Participant):
Thank you, precise definition of user roles is key for the design of access rights and the interface itself. Clear separation of functionality by user type will help both security and overall clarity of the system.
Now I would like to focus on the approval rules. In addition to the amount and type of request - are there any other parameters that govern your approval routes? For example, project number, type of financing, organizational unit, etc.?

Respondent:
Yes, in addition to the amount and type of request (e.g. purchase request, contract, invoice), other parameters also play an important role for us. Very often, the organizational unit decides - a different route applies to the technical department, another to marketing or HR. Another important parameter is the project number, because for project financing we have specific budget officers or project managers assigned who must see and approve the request. For some requests, the source of funding also matters - for example, when it comes to expenses paid from subsidies or grants, specific approval is needed from the budget controller. Sometimes the type of supplier also plays a role - if it comes to a new entity, the request automatically goes to the supplier database check. That is why I would like the new system to allow flexible combination of multiple criteria and automatically determine the approval route based on them - ideally based on pre-set rules that we could edit ourselves without the need for IT intervention.

Interviewer (Participant):
Thank you for the detailed explanation of other parameters that affect approval paths. The flexibility to combine multiple criteria to determine the approval path is really important for effectively managing different types of requests. We now have a pretty clear idea of the functionality of the new system. I have one more question – are there any specific metrics or indicators that you would like to track in the system to evaluate the efficiency and performance of the new approval system? For example, the duration of individual steps, the number of rejected requests, or the proportion of requests approved within the specified deadlines?

Respondent:
Yes, monitoring system performance and process efficiency is crucial for us – not only from an optimization perspective, but also for presentation to management. I would definitely like to have an overview of the average request approval time – both overall and by individual approval levels. This will show us where the process is stalling and who is the bottleneck. I would also like to monitor the number of rejected or returned requests and the reasons – so that we can detect recurring errors or insufficiently prepared requests. The proportion of requests processed within the specified time frame is also important – e.g. within 3 or 5 days – and for requests marked as urgent, I would like to see whether they actually went through a faster route. I would also monitor the volume of approved expenses by category, department

and project, which helps with ongoing budget management. And last but not least, I would be interested in the activity of individual users – for example, the number of requests processed, the response time of approvers, and the proportion of manually edited approval routes, which may indicate that the logic of the rules is not working somewhere. These metrics should be easily accessible in clear graphs and tables, ideally with the option to export and connect to our BI tool.

Interviewer (Participant):
What factors do you consider to be the biggest challenges in transitioning to a new approval system, especially from the perspective of users in the finance department and other teams involved? What should be key to successfully managing this transition?

Respondent:
I consider the biggest challenge to be a change in habits and resistance to change – many users are used to the current system, even if it is inefficient, and will tend to distrust the new solution or use it "the old way". The transition can be difficult, especially for older colleagues or those who are not technically proficient. In the finance department, the challenge is to align new workflows with the set control mechanisms, which today often operate "sideways" outside the system. A quality migration of data and running requirements will also be an important factor to avoid loss of information or chaos in the transition period. The key to success will be the early involvement of users in testing, quality training adapted to real scenarios and the availability of support in the first weeks of operation. It is also necessary that the new system really brings improvements – if it is just different but not better, users will not want to use it. Therefore, it must be intuitive, fast and bring tangible benefits – for example, less manual work, better overview or less email communication.

Interviewer (Participant):
I would now like to focus on the application of the new system itself. How would you envision a mobile application for the new approval system? What specific features or capabilities should the application include to be useful for approvers and other users who often work outside the office?

Respondent:
I see the mobile application as an absolutely essential tool for ensuring smooth approval, especially for managers and supervisors who often travel or work outside the office. The application should be clear, fast and, most importantly, as easy to use as possible. The home screen should have a clearly visible overview of pending requests - ideally with color-coded priorities or deadlines. For each request, there should be a preview of key data (amount, center, description, supplier), the option to view attachments (invoice, quote, contract), attached comments and the entire approval history. The user should be able to approve, reject or return the request with a comment with one click, including voice dictation of the note. The option to delegate approval directly from the mobile phone is also important - for example, in case of sudden indisposition. Push notifications for new requests or urgent requests and offline mode are a matter of course, when requests can be prepared and sent later when reconnected. The app should be available for both iOS and Android and support biometric login (e.g. fingerprint, Face ID) for convenient but secure verification. Overall, it should be built so that the approver doesn't have to "hunt for information" - everything important should be immediately at hand.

Interviewer (Participant):

Thank you for the detailed description of the requirements for the mobile application. It is clear that you place great emphasis on simplicity, speed and functionality, which are key to an efficient approval process, especially for users who are not always at their computer.
I have one more question – what is your perspective on integrating the mobile app with other tools or systems you already use in your company? For example, how should the app communicate with your ERP or accounting system regarding approved requests?

Respondent:
The integration of the mobile application with other company systems is absolutely key for me - we want mobile approval to be not just an isolated "overview", but a full-fledged part of the entire process. This means that once a user approves a request in the application, this step should be automatically transferred not only to the approval system itself, but also to the connected ERP system (e.g. Helios), where the request can create a basis for an order or reservation of a budget item. Likewise, the application should allow you to view related information from the ERP - for example, check which order an invoice belongs to or whether it is in accordance with the contract. If the user works with a DMS or has access to contract documentation, it should be possible to simply open a connected document stored elsewhere from the mobile phone. Ideally, the application would use the same API interfaces as the web system to ensure that all operations - whether via mobile or desktop - take place on the same database, without delays or inconsistencies. Integration with the company calendar (e.g. Outlook) could then help automatically display the availability of approvers or set up temporary delegations based on absences. Overall – mobile applications must be a “full-fledged player” in the entire infrastructure, not just a peripheral tool.

Interviewer (Participant):
Great, your ideas about integrating the mobile app with other systems are very clear and aim to create a strong connection between the various tools in the company. Full integration that ensures data is up-to-date and consistent is key to a smooth and efficient workflow.
Do you have any other specific requirements for security and data encryption in the mobile application that we should take into account when designing?

Respondent:
Yes, definitely – security in the mobile app is just as important to us as in the desktop version, maybe even more so, as devices can be more easily lost or compromised. First of all, I demand that all data be encrypted both in transit and in storage on the device – be it documents, request metadata or action history. Access to the app should be protected biometrically (fingerprint, Face ID) or with a strong password, with automatic logout after a certain period of inactivity. It is also important that the app does not allow downloading documents outside of a secure environment – i.e. no saving attachments to the phone’s native storage, but only viewing within the app.
I also expect remote management of the application via MDM (Mobile Device Management), so that in the event of loss or departure of an employee, the device can be remotely erased or access blocked. The application should support two-factor authentication upon first login, or when approving above a certain financial threshold. And of course, an audit trail - all actions performed in the application must be recorded in the same way as in the web system, including timestamp, IP address and user identity. I believe that these requirements will significantly contribute to the protection of sensitive data and the credibility of the system from the point of view of internal audit and compliance.

Interviewer (Participant):

These are very specific requirements for mobile app security that will definitely ensure a high level of data protection and minimize the risks associated with using mobile devices. Integrated encryption, biometric authentication, remote management and audit trail are essential to maintain high security standards. I have one more question regarding user experience – what type of support would you imagine for a mobile app? For example, should there be a help function or chat for immediate assistance if a user encounters a problem with approval on the go?

Respondent:
Yes, I would definitely expect a mobile app to offer users basic integrated help directly in the interface – ideally in the form of short descriptions, help icons for individual fields, or contextual tips on first use. New users could also be presented with an interactive guide that guides them through the basic functions of the app. For operational assistance, I would greatly appreciate an integrated chat or at least a quick support contact button that would either open a conversation with the helpdesk (for example in MS Teams, if it could be connected), or generate an email with information about the device and the user, so that they don't have to write everything down manually.
Users in the field often don't have time to search for manuals or call, so quick help in a few clicks can make the difference between them being willing to actively use the system. If technically possible, a "report a problem" feature directly from the app would also be great, where the user could report a technical issue or feedback with one click, or attach a screenshot. Overall, support needs to be accessible, fast, and mobile-friendly – that is, concise, understandable, and immediately usable.

Interviewer (Participant):
These are great suggestions for ensuring that users have easy access to the support they need and can resolve issues quickly without unnecessary delays. Integrated help, chat, and the ability to report issues directly from the app would greatly improve the user experience and streamline the entire process.
I would also like to ask what feedback methods you would like to have available for users after the new system is launched? How would you like to collect and evaluate their experiences and any suggestions for improvement?

Respondent:
Once the new system is launched, I would definitely like to have a systematic way of collecting feedback so that we can identify weaknesses and respond to user needs in a timely manner. Ideally, the system would have a built-in feedback feature directly from the user interface – a simple form with quick ratings (e.g. stars or emoticons) and optional comments. This mechanism could be discreetly available in every part of the system, or appear after a certain action is completed (e.g. after a request is approved).
At the same time, I would like to see regular short online questionnaires in the first months focused on various areas - usability, system performance, interface clarity, mobile applications, etc. It will also be key to collect feedback through key users in individual departments, who act as "ambassadors of change" - they can share their team's experiences at regular intervals and help interpret some comments in context.
It is important for me that this feedback is not just formal, but realistically evaluated – ideally in the form of a clear dashboard or report, where we can see the main topics, the frequency of comments, and possibly suggestions for improvement. Based on this, adjustments and development of the system should then be planned. If users see that their suggestions lead to specific changes, they will be much more willing to use the system and cooperate.

Interviewer (Participant):
This is a great approach to collecting feedback that will allow for active improvement of the system and will motivate users to continue using it. A rapid evaluation system, regular questionnaires, and key user engagement are effective tools for obtaining quality feedback. Last question: how would you like the entire transition process to be managed? How should it be coordinated so that all teams, departments, and users are adequately informed and prepared for the change?

Respondent:
Ideally, the entire process should be managed as a project with a clear schedule, defined milestones and responsibilities, and a central coordinator (e.g. project manager) should be provided, who will be connected to both the system supplier and individual departments. It is crucial that there is a transparent communication strategy from the beginning - users must know why the change is happening, what the new system will bring to them and when it will start affecting them. I welcome, for example, an internal microsite or Teams channel, where news, instructions, training and FAQ will be shared.
At the same time, I would recommend creating a network of key users (power users) in individual departments - they will not only test the system, but also train colleagues and provide the first level of support. We should involve end users gradually - first a pilot operation with one department, then expansion to the entire company. I would place great emphasis on practical training (ideally on real data), available support during launch (e.g. hotline or available "floorwalker") and a clear support plan after launch - i.e. what to do in case of problems, where to turn, how to escalate.
Finally, the entire transition should be continuously evaluated and tuned. This means regular reviews, collecting feedback, and planning adjustments based on real-world experience. If people feel like they are part of the process and that their voice has an impact, the adoption of the change will be much smoother.

Interviewer (Participant):
Great. Clear coordination, transparent communication and involvement of key users at each stage are essential for a successful transition to a new system. Preparation through training, support during the launch and continuous evaluation of the transition will help ensure a smooth implementation and positive acceptance among users. Thank you very much for all the detailed answers that helped me get a very clear picture of your requirements and ideas for the new system. I will definitely contact you if I have any further questions or clarifications!

Respondent:
It was a pleasure – thank you for your thoughtful and targeted questions, which allowed me to structure my thoughts well and really name what we need from the new system. I believe that if we manage to design and implement a new approval system according to these principles, it will significantly simplify our work, increase transparency and reduce the administrative burden across the entire company. I will be happy if you involve me again during the next phases – whether in testing, evaluating functionality, or creating training materials. I look forward to further cooperation!

Conclusion of the interview with participant 3

Interviewer: Thank you for the sample interview, it was very interesting. Now I would like to ask a few things about your usual practice. What development methodology do you usually use in the companies where you have worked?

Respondent: We mostly worked more traditionally – something between waterfall and a lighter version of agility. It wasn't pure Scrum or anything strict, more of a hybrid approach. We always tried to plan ahead, but at the same time be a little flexible when things changed.

Interviewer: What does it look like in practice? Do you have planned iterations or do you follow more traditional milestones?

Respondent: We usually have larger units with clearly defined milestones, but in between there are smaller iterations – prototypes or test versions, for example. So something is submitted, feedback is given, and then it is adjusted. It is not Scrum with daily stand-ups, but we try to react quickly.

Interviewer: What about your interviewing strategy? How do you approach eliciting requirements?

Respondent: It depends a lot on who I'm talking to. Users usually don't know exactly what they want, so I start more generally - "what are you doing now, what bothers you, what would you like differently". And I gradually narrow it down, for example using scenarios or specific situations.

Interviewer: And do you prepare your questions in advance, or do you rather let them flow?

Respondent: I prepare the points I want to cover – for example, topics I need to ask about. But I often adjust my questions as the conversation progresses. The main thing is not to just stay on the surface. When someone says “that’s complicated”, I ask “what exactly is complicated about that?”, “can you show me an example?” and so on.

Interviewer: And what types of requirements do you collect most often? Are they more functional requirements, or also non-functional and technical?

Respondent: We most often start with the functional ones – what should the system do, what actions should it be able to do. But I try not to forget about the non-functional ones either – for example, response speed, usability on mobile, accessibility. I collect technical requirements more indirectly – either from developers, or if I come across something that could have technical impacts, I write it down and we deal with it later with the team.

Interviewer: And do you try to verify the requirements directly during the interview?

Respondent: If it's possible, then yes. For example, we'll try to go through a screenplay or script with the respondent. I'll see if it makes sense to them, or if they start saying "but I'm also interested in this." That helps a lot.

Interviewer: And do you have any tried-and-true questions that almost always work for you?

Respondent: Definitely – like “what is most frustrating for you right now?” or “what takes up the most time?”. Then also questions like “if it were supposed to be ideal, what would it look like?” or “what goes wrong most often?”. Such questions often open up completely new things that I wouldn’t have asked otherwise.

Interviewer: How do you deal with when the user says a solution rather than a need?

Respondent: That's common. When someone says "I need a button here," I ask "what do you want that button to do?" or "what should be the result?" I try to get beyond what they're saying to why they actually want it.

Interviewer: Thank you very much, that's very helpful. And one last question - how do you deal with conflicting or unclear requirements?

Respondent: In such cases, I write it down as an open point and try to arrange another meeting as soon as possible, ideally with all parties involved. For smaller projects, it goes quickly – we just sit down and solve it together. If not, I send suggestions for solutions and ask for feedback.

Participant 4

Beginning of the interview with participant 4

Interviewer: Hello, thank you for taking the time to do this interview. Before we begin, just to relax – how are you today?

Respondent: Hello, thanks for asking, I'm doing pretty well. A bit of a hectic week, but otherwise fine. How about you?

Interviewer: Thank you too, that's fine. So let's get to it - how long have you been working as a business analyst?

Respondent: I've been working for about a year. This is my first position in this field.

Interviewer: And how did you get this role?

Respondent: I studied computer science and I was always interested in the connection between business and IT. When I finished school, I came across an offer for an analyst position at a smaller company, so I gave it a try. They were interested in someone junior to learn from, so it worked out.

Interviewer: Did you have any other job role before?

Respondent: Just part-time jobs and short internships while at school, but this role is my first "full-time".

Interviewer: And in what types of projects do you have the most experience so far?

Respondent: Mostly it was IT projects – for example, implementing new systems for clients or modifying existing solutions. Sometimes process analysis is also mixed in, but it is more of a marginal role.

Interviewer: Have you worked in different companies, or do you only have experience in one?

Respondent: So far, only from one company – it's a smaller IT consultancy.

Interviewer: Is the company small or larger?

Respondent: Small, up to three hundred people for sure. A more family style, everyone knows each other, quick decisions.

Interviewer: Great, thanks for the answers., so we can start the experiment we've prepared. Here's your assignment, and once you're ready, I'll start your respondent.

Participant-Respondent Interview (ChatGPT)

https://chatgpt.com/share/67e4564c-7f64-8008-bf46-4470eba5d22a
Interviewer (Participant):
Hello, thank you for taking the time to talk to me today about the current approval system you have in place in your department and what its successor should be able to do. In order to design a truly useful solution, I would first like to ask you. Could you describe to me what types of requests or documents you currently approve most frequently in your department?

Respondent:
Hello, of course, I am happy to provide the necessary information. In our finance department, we most often approve three types of requests: purchase requests, budgets and internal financial documents, such as proposals for transfers between budget items or extraordinary expenses. The largest volume is clearly made up of purchase requests - from regular office supplies to external services to investment purchases. For these, it is important to clearly see the budget coverage, approval stages and who is currently "on the move". With budgets, it is mainly about their proposals, changes and approval of annual or quarterly plans, where multiple levels of approval across departments play a role. Internal financial documents are less common, but often more difficult to assess, because they tend to be specific and require more thorough argumentation. Would you like me to describe in more detail the approval of purchase requests as the most common type?

Interviewer (Participant):
Yes, that would be great, let's take a closer look at purchase requisitions. What does the entire purchase requisition approval process look like currently, from submission to final approval? Can you describe the steps involved and who is typically involved?

Respondent:
Sure, I'll describe it step by step, how it currently works for us. The process begins with an employee, usually from another department, entering a purchase request into our internal approval system. They fill in the purpose, estimated price, required delivery date and attach attachments, such as offers or technical specifications. The first check is carried out at the level of the head of the department concerned - he approves the factual accuracy and necessity of the request. If everything is in order, the request goes to the finance department, where it is first checked by the budget officer, who verifies whether there are enough funds in the given item. If not, the request is returned or a budget change is consulted. If the budget is in order, the request continues to the head of the finance department - that is, to me - where I

assess the overall justification of the expenditure and approve it from the perspective of budgetary discipline and compliance with the plan. In the case of higher amounts, the company director also approves. In the system, each of these levels is recorded as one workflow step. The problem is that the system is not flexible - it is not easy to change approval routes or add an additional step, for example when a legal review is needed. It also lacks good notifications and an overview of where the request is at the moment. Are you also interested in how long it usually takes or where it most often gets stuck?

Interviewer (Participant):
Yes, that's exactly what I'd like to ask. How long does the entire purchase requisition approval process usually take - from submission to final approval? And where do the most frequent bottlenecks or delays occur?

Respondent:
Ideally, if it is a regular request with a low amount and all approvers are available, the entire process can take two to three working days. Unfortunately, this is not the rule. In reality, we are more likely to be in the range of five to ten working days, and in some cases, especially for more complex or expensive purchases, it can take longer than two weeks. The most common delays occur in two places: on the part of applicants who do not provide complete or correct documents - missing price quotes, item breakdowns or internal approval with the project sponsor. On the other hand, at the level of approvers, especially if someone is on vacation or out of the office and the system cannot redirect the request to a representative. Delays can also occur in our finance department when the budget needs to be assessed in multiple budget lines or when consultation with another department is required - but this is not the most common problem. I would also like to mention that the system does not allow for effective overview or escalation - if the request "gets stuck", we have to deal with it manually by email or by phone.

Interviewer (Participant):
Thank you, that is a very valuable description. You mentioned that the system cannot route the request to a representative if the approver is unavailable. How do you currently handle these situations - when someone is out of the office or on vacation? Do you have a fallback procedure for how the approval process works, or does the entire process stop until the person returns?

Respondent:
Yes, unfortunately, in the current system we do not have an automatic substitution function. When one of the approvers is out of the office – whether due to vacation, illness or a business trip – the process usually stops and we wait for him to return. If we know in advance that someone will be unavailable for a long time, we try to deal with it promptly – either we redirect the request manually via IT (which takes time and is not a standard procedure), or the applicant urgently notifies that intervention is needed. In extreme cases, approval is handled completely outside the system, for example by email, where the person in question writes their statement and then we in finance add it to the system “manually” so that there is a trace. This is inefficient and problematic from an audit trail perspective. Ideally, we would need the system to be able to automatically assign a representative according to the specified period of absence, or at least to have the option of delegating approval.

Interviewer (Participant):

I understand, thank you. This sounds like one of the key areas for improvement. I would now like to ask about the user interface itself and working with the system. How would you describe the clarity and intuitiveness of the current system from a user perspective? What do users find annoying or what do they find satisfying?

Respondent:
From the user's perspective, the interface of the current system is rather a weak point. It is unclear, graphically outdated and unintuitive - for example, the list of requests cannot be easily filtered by status, date or type of expense, so when someone is dealing with multiple requests, they literally have to "dig through" it. Users often do not know where their request is, who currently has it for approval or why the process has stopped - there is no workflow visualization or any delay notification. Notifications are sent, but only by email and are often not understandable or complete - for example, just a notification "request no. 358 was returned" without information about why and by whom. On the contrary, what users appreciate is that the system is simple in the sense that it has few functions - it can be learned quickly for basic use. But as soon as something more complex is needed, such as searching for approval history, pulling up an expense overview or working with a larger amount of data, it starts to become very uncomfortable. In general, the feeling of using it is that the system "somehow works", but it is more limiting than helpful.

Interviewer (Participant):
You mentioned approval history and expense overview... that brings me to another question. What reports or overviews do you currently get from the system? Are they sufficient for you, or would you need other output options, for example for analysis, audit, or business management?

Respondent:
This is a very pertinent question, because outputs and reports have long been a weak point of the current system for us. At the moment, we only have very basic overviews available - for example, a list of requests for a certain period, with their basic data such as number, applicant, amount and status. However, there are no standard outputs that would show the development of approvals, the ongoing use of budget items or, for example, an analysis of the average approval time by type of request. If we want anything more advanced - for example, a report for an audit, where a sequence of events is needed for each request, including the time of approval by individual persons - we have to request the data through IT and manually clean and supplement it in Excel. For the company's management, we often prepare our own reports combining data from multiple systems, because we do not get everything we need from the approval system. Ideally, the new system should allow data export to Excel or Power BI, have predefined and user-configurable reports and, ideally, a dashboard with key statistics in real time. I also have specific examples of reports that would be of great help to us - can I provide them?

Interviewer (Participant):
Yes, definitely, that would be very useful. What specific reports or overviews would you like to see in your department in the new system? Can you describe what they should contain and what you would use them for?

Respondent:
Certainly, I have more in mind, but I will start with the most important one – the so-called overview of budget utilization according to approved requests. This should show the volume

of funds approved in a given period, in which budget items, who the applicant and approver were, and what the remaining budget allocation is. We use it both for continuous monitoring of utilization and for preparing budget forecasts and cash flow management. Ideally, the report should allow filtering by center, period, type of expense, or request status (e.g. approved, pending, rejected). I would also greatly appreciate a time analysis of the approval process – i.e. average and median approval times according to individual workflow steps, or a notification of requests that have exceeded a certain limit. We would use this both to identify bottlenecks and for management reporting. If you are interested, I can also mention others, such as an overview of rejected requests and the reasons for their rejection.

Interviewer (Participant):
Perfect. Sure, I'm also interested in other types of reports that would make your work easier. Could you also describe the overview of rejected requests and possibly add others that you can think of?

Respondent:
Of course. The overview of rejected requests should display not only basic data (applicant, amount, date of submission), but also the reason for rejection, the stage at which the request was rejected, and who made the rejection. Similar errors are often repeated – e.g. insufficient justification, outdated offers, or unrealistic deadlines – and such a report would help us identify weak points in submitting requests. At the same time, it would be useful for training applicants or for setting up regular feedback to departments.
Another important report that we are missing is an overview of requests in process, i.e. a live list of all requests that are currently in approval, divided by status and workflow steps. This should allow not only passive monitoring, but also active management - e.g. notifications for requests that are waiting for more than X days, or the possibility of escalation.
And the last thing that comes to mind now is statistics by applicant or department - how many requests were submitted by whom, what is their average amount, what is the approved/rejected ratio. This will help us compare individual departments in terms of budgetary discipline and efficiency.
If the new system also supports custom visualizations or connections to BI tools, it would be ideal for us to be able to compile some of these views ourselves without IT intervention.

Interviewer (Participant):
Great, let's stay with integrations and connections with other tools for a moment. Are you currently using any other systems (e.g. accounting software, ERP, DMS) that the new approval system should be connected to? And if so, what data should ideally be transferred between them?

Respondent:
Yes, definitely – we have several key systems with which the new approval system should be linked. The first is our accounting and ERP system, specifically we use Helios iNuvio. Ideally, the current status of budget items should be automatically loaded from it into the approval system so that the applicant and approvers can see in real time how much money is available. Conversely, after the request is approved, the approval system should transfer information about the approved expense to the ERP – at least the amount, budget item, center and approval history – so that the expense can be either automatically booked or at least a document can be prepared in advance.
We also use a DMS system for working with documents - there we would appreciate a connection in the form of the ability to attach documents from DMS to requests or,

conversely, archive finally approved requests with attachments back to DMS under the correct folder and labels.

We also have an internal HR system, from which we would need to take information about the organizational structure and substitution - so that the new system knows who has the right to approve within which department and who is currently the representative in case of absence.

If the new system had an API or interface for data connection, we would definitely use it to connect to reporting in Power BI, as I already mentioned. Do you want me to describe the specific data flows between the approval system and the ERP in more detail?

Interviewer (Participant):
Yes, go ahead. Could you elaborate on what specific information should flow between the approval system and the ERP – both when a request is initiated and after it is approved? I am also interested in who should have access to these data transfers and whether you prefer one-way or two-way synchronization.

Respondent:
Sure, let's break it down in more detail. At the beginning of the process, when entering a request, the approval system should read the following information from the ERP: current budget items including balances, assignment to cost centers and possibly dimensions such as project, order, or expense type. Thanks to this, the requester could directly select the correct item and would be sure that the request "fits" the real budget. This would eliminate many errors and returns. This transfer can ideally be bidirectionally updated in real time (e.g. via API), but daily reading would also be acceptable.

After the request is approved, the system should automatically send the following data to the ERP: request number, approval date, final amount, selected budget item and center, name of the applicant or supplier (if already known), and also the approval status - i.e. who approved, when and in what order. This data serves either as a basis for issuing an order or for preliminary accounting of the obligation (for example, in the case of framework contracts). A return flow could then come from the ERP - for example, when an approved request is matched with an invoice, so that it is possible to monitor the implementation of the expense against the approval.

As for access, each applicant and approver should be able to read budget data within the scope of their authorization - i.e. according to the center. Writing data back into the ERP should be reserved only for approved users, typically from the finance department or IT, or it could be automated by a process with clearly defined rules. Ideally, all transmission should be logged so that the audit trail is clear.

We generally prefer two-way synchronization, as this is the only way to ensure that data is consistent and avoid unnecessary remapping and manual checks. But I understand that it can be partially one-way to start with, with the possibility of expansion.

Interviewer (Participant):
Great, thank you. Now I would like to ask about the request submission process itself. How do you perceive the comfort of submitting a request from the perspective of an ordinary user? What is the biggest obstacle or inconvenience for users in the current system when completing a request?

Respondent:
From the perspective of a regular user, entering requests in the current system is more frustrating than comfortable. The biggest problem is that the form is inflexible and does not

offer any contextual help or validation - the user often does not know exactly what to fill in, what the required formats are, or what each item means. If they fill in something incorrectly, they only find out after we return the request with a comment, which prolongs the entire process.
Another problem is that it is not possible to simply copy a request - so if someone submits a similar request regularly, they have to fill it out all over again from scratch. This is a common problem, for example, with IT or operations, who often deal with recurring orders. There is also no option to save a detailed request as a draft, so if the user interrupts work or loses their connection, everything is lost.
Another major obstacle is the restrictive work with attachments – only certain types of files can be attached, sometimes it fails due to size, and most importantly, it is not possible to meaningfully sort or comment on attached documents. This creates confusion for requests that have multiple documents (e.g. quotes, approval emails, internal calculations).
And the last thing – the user does not see in advance what the approval process will be like. That is, he does not know how many levels of approval he will have to go through, who will approve it, and when he can realistically expect the result. This reduces the motivation to use the system and increases the number of "detours" outside the system. It would be ideal if the system could automatically build an approval workflow based on the input and display it immediately.

Interviewer (Participant):
Thank you. I would now like to ask about feedback during and after the approval process. How is the company currently handling informing applicants about the progress and outcome of the approval process? And what do you think the new application should offer in this regard to make it effective and understandable?

Respondent:
Today, it is, frankly, very weak. The requester receives an email notification that the request has been submitted, and then another that it has been either approved or rejected. However, in the meantime, several steps often occur - approval by a superior, review in finance, approval by management - and the requester has no ongoing information about these. If he wants to find out where the request is, he has to log into the system and search for it manually, which is complicated, especially if he has multiple requests. Moreover, the system does not clearly show who has the request with him, so we often end up with phone calls and emails asking "can you please check if someone has already approved it?".
The new system should definitely offer ongoing notifications – ideally for every status change, with brief information about what happened (e.g. “approved by department head, awaiting funding”) and who is next in line. Visualization of the entire workflow would also be a great benefit – the applicant would see a clear timeline or status diagram, clearly indicating which steps have already been taken, who approved them, and what is currently happening.
After approval, the user should be able to view the entire approval record – including all comments, dates and names, ideally exportable as a PDF for archiving or attaching to an invoice. For rejected requests, it is essential that the reason for rejection is stated – mandatory for the approver – and the option to enter a new, modified request directly from the original. Overall, we want the applicant not to have to "hunt" for information, but for the system to communicate clearly on its own.

Interviewer (Participant):

That makes sense. I'll now move on to the approval area from the perspective of managers. What features or information do you think the new system should offer specifically to approvers - to make their work easier and make decisions faster?

Respondent:
From the approvers' perspective, it is crucial that they have quick and clear access to all relevant information at a glance - without having to click through to different sections or search for connections. Ideally, each approver would see a list of requests awaiting their decision on their home screen, with a brief summary for each: applicant, amount, type of expenditure, budget item, budget status (e.g. "available funds: CZK 120,000"), and possibly a brief justification or attachments.
Furthermore, the system should allow for quick approval or rejection directly from the overview, without the need to open each request separately, for simpler cases. For more complex cases, there should be easy access to attachments, approval history, comments from other approvers, or perhaps comparison with similar requests.
Delegation is also important - so that the manager can set up a representative, or the system can automatically transfer authority during his absence. In addition, there should be an overview of the status of all requests from his department, not just those waiting for his approval - so that the manager has control over what is happening in his team.
And last but not least - notifications should be intelligent, e.g. a summary of all new requests once a day, the ability to set filters by amount or type of expense, and notifications for requests that are pending for more than X days. In short, a system that doesn't annoy, but helps.

Interviewer (Participant):
Great, I understand. Now I would like to ask about specific situations. Are there cases where approval is needed outside the standard workflow – for example, ad hoc involvement of the legal department, project management or an external person? And how do you solve this today?

Respondent:
Yes, such situations occur quite often and today we deal with them quite uncomfortably, because our system does not allow flexible workflow adjustments. A typical case is, for example, the purchase of a service or software, where a legal assessment of the contract is needed - in an ideal world, a lawyer would be part of the workflow, but today we deal with it outside the system. The request is approved in parallel, and at the same time the contract is sent to the lawyer by email. As soon as he comments on it, the note is attached or forwarded - without a record in the system. It works similarly with project expenses, when the input of the project manager is needed, but he is not set up in the approval route, so we secure his approval again separately.
Sometimes there is a requirement where someone outside our company has to approve – typically for joint projects with partners or a superior body within the group. There is no way to formally involve them in the process, so all approvals are done manually, via emails, and we just “enter” it into the system to keep a track of it – which is of course not ideal in terms of transparency or auditability.
In the future, it would be ideal if the system allowed for ad hoc addition of an approver or a parallel step, or approval conditional on comments from the legal or project department. As well as the ability to add an external user with limited access who could approve or comment on a specific document without having access to the entire system.

Interviewer (Participant):
Thank you, a great area to cover. Now let's move on to the audit and legal aspects. What are your requirements for an audit trail in an approval system? What do you think the system should record, and how should these records be available for checks or audits?

Respondent:
An audit trail is absolutely essential for us, mainly for internal and external audits and compliance with legislation such as the Accounting Act, the Financial Control Act or GDPR. The system should record all key events in detail, including: who created the request, when and exactly what was filled in, who approved or rejected it in what order, including the date and time of each step, and possibly a comment, if any. It is also important to record every change to the request - for example, adjusting the amount, adding attachments, changing the workflow or delegation - including who made it.
Records should be immutable and traceable even after several years, ideally with the possibility of exporting to a readable format (PDF, Excel) or accessing via a secure interface. For audits, it is also important to be able to generate a complete history of a specific requirement – the so-called audit trail – including all versions and attachments, so that it is clear what the decision-making process was and who was responsible for which step. Another requirement is the ability to filter and search audit records according to various criteria – for example, all requests approved by a certain person, rejected in a certain period or modified after approval. It is also important that audit records are available for defined roles (e.g. internal control, auditor), but at the same time protected against unauthorized access – i.e. correctly set permissions and logging of access to this data. In short, we want to be sure that everything that happens in the approval system can be proven and documented, and at the same time that only those who have the right to do so have access to this information.

Interviewer (Participant):
I understand, thank you. Now I would like to ask about security and access rights management. What permission levels do you think the new system should support? And what rules should apply to role management, access to requests and sensitive data?

Respondent:
From a security and authorization perspective, we would like the new system to support multi-level role management, both by function (e.g. applicant, approver, financial controller, administrator) and by organizational structure (e.g. access only to a center, department, or project). Each user should have access only to those requirements and information that relate to their competence - i.e., a department head can see all the requirements of his team, but not the requirements of other departments.
It is important for approvers and reviewers to have different permissions for editing and approval, and for the system to clearly distinguish who can only view, who can comment, and who can approve or reject. In finance, we need to be able to edit the budget allocation of a request, but not, for example, change its content without the requester's consent - the system should record such interventions and limit them by role.
It is also essential to protect sensitive data – for example, requests related to HR services, legal documents or investment expenditures – where access should be limited to selected individuals. Ideally, the system should allow setting the sensitivity of the request and dynamically adjust access rights accordingly.
When it comes to role management, it should be possible to manage them centrally, but also have a certain degree of delegation - for example, so that a manager can set up a

representative himself, or add a colleague to the approval process during his absence, without having to intervene in the system through IT. And of course, all changes to permissions should be logged and available for audit.
We also prefer connection to corporate Active Directory or SSO to enable centralized identity management and reduce the risk of access abuse.

Interviewer (Participant):
Great. Now let's move on to user support and adoption. How do you think the new system should support users during use? Are you interested in integrated help, a guide, a chatbot, or something else? And what forms of training and support would you prefer during its implementation?

Respondent:
Yes, user support will be key when implementing the new system, especially since our colleagues have different levels of digital literacy and some are sensitive to system changes. We would certainly appreciate if the new system included integrated context-sensitive help - i.e. a small icon next to each field or step explaining what to fill in and why. It would be ideal if the system could adapt to the type of user - i.e. the applicant would see a simpler interface, while the approver would see more information.
An interactive guide when using the system for the first time or when changing functionality would also be a great benefit - that is, the system "guides" the user step by step with simple instructions. A chatbot or integrated search in the help would certainly be useful for frequent questions, but only if it is well-designed - the user is often frustrated if the answers do not correspond to reality.
Regarding support for system implementation, we would prefer a combination of training and stand-alone materials. That is, short online training for different roles – especially for applicants, approvers and administrators – ideally in the form of videos or webinars that anyone can watch at any time. At the same time, we would like brief instructions and checklists in PDF format or presentations that people can print out. For more complex functions, we would appreciate the possibility of individual consultations or support via a helpdesk, at least in the first months.
It is important that the training is not just about technical control, but also about understanding the logic of the approval process - i.e. why certain information needs to be filled in, what happens after approval, etc. This significantly reduces resistance to change.

Interviewer (Participant):
Adoption support is often underestimated. I would now like to ask about the future. What other features or capabilities could you imagine as useful in the future, perhaps even beyond current needs? Do you have a vision for where the system could continue to develop?

Respondent:
Yes, absolutely – I think once we build a robust foundation for efficient and transparent approvals, we can gradually add features that will make it a truly intelligent tool for managing expenses and budgets. For example, in the future, I could imagine the system actively analyzing approved requests and providing recommendations – e.g., highlighting recurring expenses that could be merged into a framework contract, or suggesting an alternative based on history.
Another interesting option is to automate approval of low-budget and routine expenses if they meet certain predefined criteria - which would relieve approvers and speed up the entire process. At the same time, the system could be able to model the impact of requirements on

the future budget - for example, show how the approval of a certain item will affect spending in the following months or quarters.
I would also like to see more advanced integration with purchasing and contracting processes – meaning that a request could directly generate an order, link to a supplier, attach the relevant contract and monitor its fulfillment. And in the long term, it would make sense to connect the system with spending trend analysis so that we can look for savings, evaluate efficiency and move corporate finances towards predictive management.
From a user perspective, I would also like a mobile app or at least a responsive web interface to allow approval from a mobile device, which is especially useful for managers outside the office. And in the future, perhaps support for voice input or integration into tools like MS Teams - but that's really a vision for the next phase.

Interviewer (Participant):
That's a very strong vision strategically. Thank you very much. I'd like to come back to the topic of mobile access that you mentioned. Could you describe to me in what situations it would be useful to be able to approve requests via mobile or tablet? And what features would you consider necessary in the mobile version of the system?

Respondent:
Certainly, we see mobile access as a very practical addition, especially for executives and top management who travel frequently or are not regularly at their computer. A typical situation is when a manager is on a business trip, has a few minutes between meetings and needs to "unblock" a request that would otherwise delay a project or purchase. The same is true when on vacation – not everyone wants or can turn on their laptop, but they always have their mobile phone with them.
In the mobile version, I would consider it necessary to:
quick overview of requests awaiting approval, ideally with the option of a simple filter (e.g. by amount or type),
a brief summary of key information for each request – applicant, amount, description, budget line, budget status and attached annexes,
the ability to view attachments (PDF, Excel, images), ideally without the need to download,
approval, rejection and comment with one click, including proposed changes,
mobile notifications (push messages) that alert you to new requests or deadline-critical approvals.
The mobile version should be clear, simple and, most importantly, fast – no one wants to fill out long forms on their phone. We don't want the entire process to be done via mobile, but approving and tracking the status of requests is an area where a mobile approach would significantly increase efficiency and eliminate unnecessary delays.

Interviewer (Participant):
Great. Now I would like to go back to the topic of attachments. What types of documents do you typically attach to requests and how do you think the new system should ideally work with attachments in terms of formats, management, clarity, or connections to other systems?

Respondent:
Attachments are an integral part of most requirements and often contain key information for approval, so their management must be significantly better in the new system than today. Most often, we attach price quotes, technical specifications, contracts, internal calculations (e.g. in Excel), correspondence with suppliers (e.g. emails in PDF), and sometimes even visualizations or project documentation.

The system should support a wide range of formats – at least PDF, DOCX, XLSX, images (PNG, JPG), ZIP archives and possibly EML (for e-mail communication). It is very important to clearly display all attachments to the request – ideally divided by type or purpose (e.g. "offers", "contracts", "internal documents"), with the option to add a note to each attachment (e.g. "version A - valid until 31 March"). Attachments should be viewable directly in the system, without the need to download, and ideally also versioned – so if someone uploads an updated offer, the system will retain the original one.

From a management perspective, quick transfer of attachments from the DMS should be enabled - either connecting directly from the document system (e.g. via a link or interface), or automatically archiving final versions back to the DMS to avoid duplication and loss.

It is also important to control permissions for attachments - for example, so that legal documentation is visible only to relevant roles, or that sensitive documents are not visible to every approver. Finally, when exporting a requirement (e.g. for audit or archiving), a package of all attachments should be part of the output, ideally with a clear indication of which requirement they belong to.

Overall, we want attachments to be not just "something extra", but a full-fledged part of the requirement - easy to find, read and securely stored.

Interviewer (Participant):
Thank you. Now I would like to focus on supporting different types of requests. Are you considering that the new system could include templates for different types of requests, e.g. purchase of services, investments, travel expenses, etc.? What types of templates would make sense to you and what should they contain?

Respondent:
Yes, we are definitely considering templates and we consider them a very useful element that would help simplify entry, reduce error rates, and at the same time ensure that requests have a uniform structure and all the necessary information. Ideally, the user would initially choose the type of request according to its purpose - and the system would load the corresponding template with preset fields, workflow, and attachment types accordingly.

These basic templates should definitely exist for us:

Purchase of common goods or services – typically office supplies, software, training. It would contain fields such as description, estimated price, supplier, budget code, required delivery date, bid attachments.

Investment expenses – e.g. IT equipment, machinery, building renovations. Here the template should contain additional fields for planned useful life, depreciation group, project number, internal justification of return, etc.

Travel expenses – used, for example, to plan a trip abroad. It would contain fields for the location, purpose of the trip, expected costs for transportation, accommodation, meals, and approval from HR or a direct supervisor.

External services with legal review – typically consulting, contractual services. Here, the template would automatically involve the legal department in the workflow and would include fields for the supplier, contract draft, service value, and performance period.

Transfer between budget items (internal request) – used, for example, when rescheduling expenses during the year. This would be a simple form stating the original and new items, amount, justification and approval from the budget officer.

Each template should not only have its own set of fields, but also be linked to a specific approval circle and required attachment types. The system should allow the administrator to continuously modify the template – for example, add fields or modify the workflow – without developer intervention. This would allow us to respond to changes in company processes or legislation.

Interviewer (Participant):
Great. Now I'm interested in seeing how multiple people collaborate on a request. Are there situations where multiple people collaborate on a request, such as a requester with a colleague, project manager, or specialist? And if so, how should the new system support collaboration when creating a request?

Respondent:
Yes, such situations occur quite often, especially with more complex requests – typically in the case of project purchases, investments, or technically specific services, where the requester submits the request formally, but prepares the content in collaboration with a colleague, e.g. a project manager, technician, or someone from IT. In the current system, we solve this by having people send the form in advance by email or collaborate in Excel, and then someone manually transfers the completed request to the system – which is inefficient and often results in errors or outdated versions.
The new system should make this collaboration much easier. Ideally, a request could be created in a conceptual mode, to which the requester could invite other people – i.e. set up sharing with a colleague who can add or comment on individual fields, attach attachments or add notes. Everyone should have a clearly defined role – e.g. editing, commenting, reading only – and the entire preparation process should be versioned, so that it is possible to return to older versions or track who changed what.
It would also be useful to have the option of internal communication directly within the request – i.e. comments or notes "inside" the form, without having to deal with things separately via emails. Ideally, with the option to tag a colleague (e.g. @j.novak) and trigger a notification.
This would allow for true collaboration on a single request without having to constantly sync versions between different tools. It would also significantly improve the quality of requests before they are sent for approval, reducing the number of requests that are returned or rejected.

Interviewer (Participant):
That's right. Collaboration in requirements preparation is often neglected. Let's stay with the preliminary phase. How important would it be for you to be able to save a work-in-progress requirement as a draft? And what features would you appreciate within drafts, such as versioning, reminders, sharing?

Respondent:
I would consider the option to save an in-progress request as a draft to be very important – in practice, situations often arise when an applicant starts filling out a request, but first needs to find prices, agree with a colleague, or obtain additional documents. Today, when the system does not support drafts, this is a major weakness – the user either has to fill everything in at once (and risks losing data), or prepares it in Excel and then rewrites it.
The draft should be saved automatically and manually, ideally with the ability to return to the last saved version. I would also like versioning of the draft - that is, the ability to save several variants, for example with different amounts or offer variants. It is important that the draft can be named or marked with a note so that the user can easily remember what it was about later.

Another key feature is sharing a draft with colleagues, as we discussed - the ability to invite someone to collaborate before sending it for approval, including comments or suggested changes. It would also be very practical to have the option to set a reminder, such as "Notify me in 3 days if the draft is not submitted", or to display an overview of unsent drafts on the main screen.
And finally, it would be great to be able to convert a concept into a request, duplicate it, or create a new request from it - this is especially practical for recurring purchases or similar requests.
Overall, this would significantly increase the comfort of work and allow for the preparation of higher quality requirements without time pressure.

Interviewer (Participant):
Great, thanks. Now let's move on to the user experience requirements. What features do you think a modern and friendly user interface for the new approval system should have? For example, is appearance, clarity, responsiveness, or adaptation to different types of users important to you?

Respondent:
Yes, the user experience is very important to us - given that the system will be used by a wide range of people across the company, from administrative staff to management, it must be really clear, modern and understandable even for less technically savvy users. The basis is a clean, visually light design, where important information is visible immediately and the user is not overwhelmed by unnecessary elements. A big plus would be responsiveness, i.e. full functionality even on a tablet or mobile phone - ideally without the need to install the application.
The user interface should be contextual and role-based, i.e. adapt to who is logging in – an applicant will see different functions than an approver, and an administrator will see different functions than a regular user. This helps with orientation and speeds up work. Consistency of control is also important – i.e. all forms, tables or actions work the same across the system.
I appreciate when the system can work with a dashboard or home screen that shows the user everything important in one place - for example, pending requests, drafts, notifications or quick links to frequently used templates. Functions such as searching and filtering requests must be simple and fast, ideally with the ability to save favorite filters or reports.
Of course, appearance isn't everything, but if a system looks outdated, it also affects user trust - a modern, clear interface helps users "believe" in the system and make it easier to accept. And this is key to successful adoption.

Interviewer (Participant):
That's a great summary. Now I would like to ask about the area of change management and system evolution. How do you think the new system should be managed and developed after deployment? Do you have an idea of who should be responsible for management, how change requests should be collected, and how often the system should be updated?

Respondent:
Yes, it is definitely important to think about what will happen after deployment. In practice, it often turns out that some needs only emerge during use, or that processes in the company change their logic over time. The new system should therefore be designed to be sustainable and flexible in the long term, and its administration should be centrally coordinated, but at the same time open to feedback from operations.

A clearly defined owner should be responsible for system management – ideally someone from operations support or IT who knows the technical side of the system but also works closely with key users (e.g. finance, HR, purchasing). In our case, it would be ideal for key roles to have their own "system representatives" who will collect input from individual departments – i.e. decentralized requirements collection but central change management.
Change requests should be collected systematically – ideally directly in the system using a feedback form or a simple "Suggest Improvement" button. Each suggestion should be recorded, assessed for impact, and included in the development plan. We would certainly appreciate a public changelog or overview of planned changes so that users can see if their suggestions are not being met.
The frequency of updates should be regular but reasonable – for example, once a quarter for minor functional improvements, and once a year for larger versions with new modules. All changes should be documented and accompanied by brief training or instructions so that users know what is changing.
At the same time, we would like to be able to make some changes to the system ourselves – e.g. editing templates, managing workflows, adding approvers – without having to contact developers. This will allow us to respond quickly and the system will not become obsolete immediately after deployment.

Interviewer (Participant):
Excellent, thank you. That concludes my set of questions and thank you very much for your openness. Is there anything else that we haven't mentioned that you would consider important for the design of the new approval system?

Respondent:
Thank you for a very well-conducted interview – you really covered the vast majority of important areas. I would just like to add one thing that is often overlooked in system design: transparency and user trust. A new system should be built in such a way that users understand what is happening with their request, why and who is making decisions about it. When a system communicates clearly, allows traceability and offers support at every step, its adoption across the company increases significantly.
At the same time, I would like to emphasize that for us the new system is not just a "tool for approving expenses", but part of a culture of responsible management. It helps us maintain budgetary discipline, manage expenses efficiently and support transparent decision-making – and if designed correctly, it can be a real partner in everyday work, not just an obligation.
If you need to add to or elaborate on any of the areas, I will be happy to continue to be available. Thanks again.

Interviewer (Participant):
Thank you very much, this was an absolutely excellent ending to the interview.

Respondent:
Thank you, that's great to hear - I appreciate it. If you need anything to add or consult on the design, testing or implementation of the new system, I'd be happy to get back in touch. Good luck with the further progress of the project!

Conclusion of the interview with participant 4

Interviewer: Thank you for showing me how you conduct an interview a moment ago. It was very helpful. Now I would like to go back to some areas of your work. I will start with the methodology – what type of development methodology do you use in your company?

Respondent: We mostly work in agile mode, specifically according to the principles of SCRUM, although adapted to our conditions. We call it “agile foundation, pragmatic practice”. We have sprints, stand-ups, retrospectives, but we do some things more freely.

Interviewer: What does this specifically mean in practice? For example, in terms of planning or backlog?

Respondent: We have a backlog, but it is not kept strictly. Rather, we have a shared board where tasks are divided by teams and types of work. When planning sprints, we try to take into account not only new requirements, but also technical debt and maintenance. It often happens that something urgent comes up during a sprint - we try to be flexible, but not at the expense of quality.

Interviewer: And as for the requirements gathering itself – what is your strategy during the interviews to make the requirements as precise as possible?

Respondent: I mainly focus on context. I try to understand why the user needs something, not just what they specifically want. I use a lot of open-ended questions like “What are you trying to achieve?” or “What would happen if this wasn’t implemented?” Then I move on to specific scenarios – like “Imagine having this new system in front of you, what would you see on the screen?” This often helps to uncover unspoken needs.

Interviewer: I'm interested in how you ask questions when you need to get technical or non-functional requirements, not just functional ones.

Respondent: It's harder with non-functional requirements because users don't often talk about them themselves. So I guide them in a targeted manner. For example, I ask: "How fast do you think it should work?" or "Who all will have access to it and from what devices?" This opens up the topic of performance, availability or security. Sometimes I also use extreme scenarios, such as: "What if your system goes down on Friday afternoon - what does that mean for you?" This often shows how important availability or support is to them.

Interviewer: Do you have a specific structure for preparing these questions, or is it more intuitive?

Respondent: It's about half and half. The basic outline is there – I start with the context, then I go through the processes and user roles, then scenarios, exceptions, reports and finally security and integration. But I react a lot to what the respondent says. If I see that he starts talking about something important aside, I let him talk and then I go back to the outline.

Interviewer: And what types of requirements do you typically collect? Do you differentiate them somehow internally?

Respondent: Definitely. The basis is the functional requirements – what the system should be able to do, what the user should see or do. Then the non-functional ones, mainly performance, security, access rights. We usually define the technical requirements together with the developers – for example, what APIs will be needed, what the solution architecture will be, data flows. In addition, I try to capture business rules – for example, restrictions from legislation or company policy.

Interviewer: When you get the requirements, do you validate them somehow? Like validation with the respondent?

Respondent: Yes, definitely. After the interview, I put together an overview – sometimes in the form of user stories, sometimes as a list of requirements. Then I send it back for approval or comments. Sometimes I present it in the form of storyboards in Miro or a clickable prototype – that tends to be the most effective because people respond better to something visual.

Interviewer: And have you ever encountered a problem where a user said something that they later changed their mind about, or it turned out that they had worded their request incorrectly?

Respondent: Sure, it happens. But that's why I try to work iteratively - I don't take anything for granted after one conversation. The key is to have room for feedback and be ready to change the specification. It also helps a lot when it's not a one-way communication - users are more engaged when they see that their comments are being incorporated.

Interviewer: Thank you, this is very helpful. Is there anything you would like to add about how you work with requirements or what you consider important in analytical work?

Respondent: Maybe just that a good analyst has to be a bit of a psychologist. It's not enough to just ask questions - you need to perceive what people don't say out loud, but it's important to them. And also have patience - many things only become clear with time or when working with other people.

Participant 5

Beginning of the interview with participant 5

Interviewer: Hello, thank you for taking the time to talk to me. Before we begin, how are you?

Respondent: Hello, thanks for asking, I'm fine. How about you?

Interviewer: Okay, thank you. I would like to ask you first — how long have you been working as a business analyst?

Respondent: I have been working as a business analyst for two years, this is my first experience in this role.

Interviewer: And how did you get that role?

Respondent: Well, even in school I enjoyed connecting technical things with what the company really needed. After school I joined a smaller company where I had the opportunity to try it out in practice. At first I helped with analysis and gradually moved to the position of a full-time business analyst.

Interviewer: Did you have any other position before?

Respondent: No, this is my first position after school. Before that, I had some short-term jobs, but nothing in the field.

Interviewer: And in what types of projects do you have the most experience so far? More likely IT, process analysis or strategy?

Respondent: I've worked mostly on smaller IT projects - like streamlining internal systems or developing simple custom tools. Here and there, some process work, but so far mostly IT.

Interviewer: Have you worked in different companies during those two years, or do you have experience from only one organization?

Respondent: Just one. It's a small company, but quite dynamic, so I had the opportunity to try out more things.

Interviewer: So you currently work in a small company?

Respondent: Yeah, it's a small team, we have about ten, fifteen people. So it's a pretty informal and very flexible environment.

Interviewer: Thanks, so we can get started on the experiment we've prepared. Here's your assignment, and once you're ready, we can begin.

Participant-Respondent Interview (ChatGPT)

https://chatgpt.com/share/67e466f0-961c-8008-bc02-29d83531af79
Interviewer (Participant):
Hello, thank you for taking the time. I would like to talk to you today about your experiences with the current approval system and your expectations for its successor. I will start with a basic question: what types of approval processes do you currently use most often in your department?

Respondent:
Hello, thank you for taking the time to do this. In our department, we most often approve purchase requests, budgets and their changes, internal transfers of budget items between centers, and of course various types of contracts - especially supplier and leasing contracts. Each of these processes has a slightly different logic and requires a different level of approval, but we handle all of these types through our internal system. I can describe the purchase request approval process first, because it is the most frequent in our company.

Interviewer (Participant):
Thank you, that would be great. Let's definitely start with purchase requisitions. So could you please describe to me step by step how the approval of a purchase requisition currently works in the current system? Who enters it, who approves it, what information is attached to it and what happens after approval?

Respondent:
Sure, I'll describe it step by step, how it currently works for us. A purchase request is typically submitted by the head of a specific center or his representative - that is, a person who has an overview of the needs of his team. In the form in the system, he fills in basic information such as the name of the requested purchase, amount, center, type of expense (e.g. CAPEX vs. OPEX), reason for purchase and, if necessary, attaches a price quote or other additional document. A major limitation is that the form is relatively rigid and we often miss some fields - for example, the ability to enter multiple price quote variants or a proposed supplier framework.
After entering the request, the approval workflow automatically starts according to a predefined scheme. First, the request is approved by the head of the superior department, then it goes to me or my colleagues in the finance department, where we check the budget coverage and the correctness of the classification. If everything is in order, we approve it. If not, we have to manually contact the client or approver before us, because the system cannot effectively deal with comments - for example, it is not possible to simply return the request for revision with a comment. Finally, if it concerns higher amounts, the request must still be approved by the management or director.
After final approval, the system should generate an order or purchase order, but this often happens outside the system – via email or Excel, because the current system does not have a connection to the purchasing module or ERP. This is a big weakness for us, because continuity and traceability are lost. So the whole process is very manual, spread out across multiple channels and often slows down due to the lack of flexibility of the system.

Interviewer (Participant):
Thank you. Now let's focus on the first part of the request form itself. You said that the form is too rigid and that some fields are missing. Could you please specifically list what additional fields or functions you would like to see in the request form so that it better reflects the reality of your requirements?

Respondent:
Yes, definitely. There are more, but I'll mention the most important ones now. The first thing we really miss is the ability to enter multiple quotes from different suppliers - currently we have to upload these quotes as a single PDF file, which complicates comparison and transparency of selection. It would be ideal if quotes could be entered in a structured way, for example as a table directly in the system, including suppliers, prices, delivery dates and notes.
The second thing we are missing is a field for the proposed supplier with a reference to a framework agreement or a central partner - now we solve this manually with text in a note. However, for us as finance, it is important to know whether the purchase falls under centrally negotiated conditions or whether it is an individual agreement.
And the third thing is the ability to mark whether it is an urgent request. The current system has one queue, without any prioritization, so everything waits the same amount of time, even if it is a repair of a critical device.
There's more - like automatic checks against budget at the time of entry or the ability to add an internal note for approvers only - but I would definitely call these three the biggest gaps in the current form. Do you want me to describe others?

Interviewer (Participant):
This is absolutely great, thank you. So multiple offers, connection to framework agreements and priority distinction. I would definitely like to get to the other add-ons you mentioned, but

before we go any further, I have one more question about the multiple offers. How specifically do you imagine the process of comparing those offers in the system? In your opinion, should the system just display the comparison, or should it also evaluate the best option based on the specified criteria, e.g. lowest price, fastest delivery?

Respondent:
Excellent question – of course, it would be ideal if the system could not only display offers, but also basically compare them according to pre-set criteria. I don't expect the system to decide for us, but it should make it much easier to navigate and give us clear inputs for making decisions. I imagine it this way: the contracting authority indicates what the main criteria are for a given purchase – for example, the lowest price, the shortest delivery time, or a combination of several factors – and the system then automatically highlights which offer best meets the specified parameters. It would help if each offer had standardized fields such as unit price, total price, delivery date, warranty length, delivery conditions, etc.
On the other hand, I don't want the system to make decisions "hard". The final choice should always remain with the user, who must also justify it - and this option to justify the choice should also be mandatory there, unless the contracting authority chooses the offer that was "best" according to the system. This is important from the point of view of internal control and audit. So comparison is possible, highlighting the recommended option is also fine, but the decision should remain with the person.

Interviewer (Participant):
Perfect, that makes sense, comparing as a basis, not as an automated decision. Now I would move on to the approval workflow. You mentioned that it often happens that you have to contact the client or previous approver manually for comments and that the system does not know how to return requests for rework effectively. What do you think the mechanism for commenting or returning a request in the new system should ideally look like? What should it be able to do so that it is practical and does not delay the entire process?

Respondent:
Ideally, the new system should allow any approver to add a comment to a request, either as a "for information" note or as a reminder that pauses the request and returns it for rework. It is important to be clear who entered the comment, what specifically they are requesting to be changed, and that this is reflected directly in the system – not just via email or outside of it. The requester should be able to edit the request and resubmit it to the approval wheel, with the system remembering the history including what was changed and why.
The comment should also include the option to mark the specific part of the form that the comment concerns – for example, "incorrectly selected cost type" or "unclear description of the purpose of the purchase". This will avoid unnecessary arguments. And ideally, the system would notify all affected persons automatically, including notification via email or other internal communication tools.
It is essential that the return does not mean a complete restart of the workflow, but that the request continues where it left off - that is, back to the approver who last commented on it. This will significantly speed up the entire process and prevent unnecessary loops.

Interviewer (Participant):
Great. So the ability to add a comment, type of comment, mark a specific field, automatic notifications and maintain context and progress in the process. I would now like to ask about the approval phase from a financial control perspective. You said that you and your colleagues check the budget coverage and the correctness of the classification. What specific

information or tools should the new system offer you so that you can perform this control more effectively than today?

Respondent:
That's right - our main task in the approval process is to verify whether the request has budget coverage, whether it is correctly classified by cost type (e.g. CAPEX vs. OPEX, investment vs. operating expense), and whether it complies with company rules. Today, we have to find most of this information manually - in Excel reports, a separate budget system, or by asking the contracting authorities.
Ideally, the new system should be directly connected to budget data – in real time or with minimal delay – and when opening a request, it should display an overview of the relevant budget: planned amount, disbursement, balance, at the level of a specific budget line and center. It would also be useful to highlight in color if the request exceeds the limit or goes into reserve.
I would also appreciate validations that would alert you to inconsistencies when you enter a request – for example, if the selected expense type conflicts with a budget line item, or if the investment code is missing from a CAPEX request. These checks would significantly reduce the number of errors and unnecessary approval rounds.
Finally, it would be great to be able to add our “review notes” directly into the system for other approvers, including management, to see. This will increase transparency and save time, as these days we often explain things side by side over the phone or via email.

Interviewer (Participant):
I understand, this is really important. Direct connection to budget data, automatic validations and the ability to record comments directly into the system. I will follow up on one specific thing you mentioned, validations and checks when entering a request. In addition to those you have already described, e.g. CAPEX without an investment code, budget overruns, can you think of any other type of check or smart watchdog that the system could perform that would make your work easier or increase the quality of the assignment?

Respondent:
Yes, of course, there are several other types of checks that would really help us. For example, it would be useful if the system monitored duplicate requests - that is, if a request with a very similar description, amount and center is submitted in a short time, it could display a warning when it is submitted, saying: "A similar request was submitted X days ago - please check if it is a duplicate." This would prevent unwanted approval of the same thing twice.
Another smart control could be to link to a code list of prohibited or restricted types of expenses – for example, there are certain costs that can only be approved by management, or are completely prohibited without special permission (e.g. entertainment above a certain limit). If the client accidentally includes this, the system should display a warning and prevent continuation without additional approval or note.
We would also like to check for VAT compliance – for example, if the amount includes VAT but the purchase is from abroad where VAT is not charged, or if the contracting authority fails to indicate that the purchase is subject to the reverse-charge regime. This currently has to be checked manually, which is time-consuming.
And the last thing that comes to mind is checking attachments – the system might require a mandatory attachment for a certain type of purchase (e.g. offer, calculation, approved project), otherwise it won't allow the request to be saved. It often happens that documents are missing and the whole process is unnecessarily delayed.

Interviewer (Participant):
Great. Now let's look at the approval process itself from the perspective of the schema. You mentioned that the workflow is predefined according to the schema and can include various levels, including management. What do you think the ideal approval workflow model in the new system should look like? Should it be fixed, or should it dynamically adapt to certain parameters of the request, e.g. amount, type of expense, center, type of contract?

Respondent:
It should definitely be adaptable – a fixed workflow is one of the problems of the current system. The ideal model should be dynamic and governed by a set of clearly defined rules, according to which the system automatically chooses the right approval path. For example, approval levels would be added or removed depending on the amount – up to 50 thousand, only the head of the department and finance approves, above 100 thousand, even the director or CFO, above half a million, perhaps even the CEO.
Likewise, the type of expense – for CAPEX, the workflow should automatically include investment project approvers or asset managers. For operating costs, on the other hand, approval may be simpler. The center is also important – some have a special regime (e.g. IT or HR), where there are also internal controls or central approvers.
And the last thing is the type of request - regular purchase, leasing, contract, business trip - each of these types can have a different approval logic, and it would be great if the system could automatically suggest the appropriate workflow based on the type selected.
At the same time, I would like the workflow to be clear for users - i.e., for the client and approvers to always see who is next, who has already approved, and who is still to go. And if necessary, there should be the possibility to manually edit the scheme in exceptional cases, of course with the appropriate permissions and a recorded history of changes.

Interviewer (Participant):
Thank you, this is a type of flexible and transparent workflow. Dynamics according to the amount, type of expense, center, request, supplemented by the possibility of exceptional adjustments and an overview of the status. I would now like to ask about integrations. You mentioned that the current system is not connected to the ERP or purchasing module and that this is losing continuity. Which specific systems or data sources should the new approval system be able to connect, and what should be done automatically as part of these integrations?

Respondent:
Yes, integration is key for us – without it, even the best approval system will still be only a half-solution. The new system should be connected primarily to our ERP system, where we record budgets, accounting structure and subsequently realized expenses. This means that ideally, when entering a request, it should automatically draw budget data from the ERP – plan, drawdown, balance – and after final approval, it should either create a purchase order directly in the ERP, or at least generate approved documents for its issuance.
It should also be connected to a purchasing portal or catalog, if we use them - this will allow you to directly select items from preset offers and speed up processing. For contractual requirements, it would be useful to connect to our contract management system so that an approved requirement can be easily converted into a draft contract, or vice versa, so that the system recognizes that the requirement relates to an existing contract.
We also need a connection to the HR system so that the system knows who has what permissions, who is the manager of whom, and who has the right to approve certain types of

expenses - this would greatly simplify workflow management and its updating when changes occur in the organization.
And last but not least, I would also appreciate integration with a DMS (document management system), where attachments and approved requests would be automatically saved - ideally with document versioning. Automating these connections should minimize manual data rewriting and document transfer outside the system - this is the main source of errors and delays today.

Interviewer (Participant):
Thank you, this is very robust. I'll stick with integrations, but from a different user perspective. What types of alerts or notifications do you think should be available in the approval system so that users (e.g. approvers or requesters) don't forget to respond and are well informed about the status of their requests?

Respondent:
This is a big topic, because today we often encounter people forgetting to approve or not even knowing that something is waiting for them - and the whole process is stalled. The minimum basis should be an automatic email notification when the request is forwarded to the user for approval. But it should not stop at just one notification - ideally, the system should also send reminders if the user does not respond for, for example, 2 or 3 days. Everyone could adjust the frequency or interval, but the basic reminder logic should be preset.
I would also definitely welcome notifications via company communication tools – in our case, Microsoft Teams – even as short messages like “You have a new request for approval” or “A request with a comment has been returned to you.” Such notifications are often more effective than email because they don't get lost in an overloaded inbox.
From the perspective of the client, the system should be able to send information about every movement of the request - who approved it, who returned it with a comment, where it is currently located. And the user should be able to set how detailed the notifications are - whether they want to receive everything or just the key moments.
And it would be ideal to have a clear "message board" after logging into the system - something like a dashboard with an overview of all my active requests (waiting for my approval, submitted by me, returned for revision, etc.) - so that I don't have to search for anything or filter it complicatedly.

Interviewer (Participant):
This is a completely understandable and practical approach of combining several tools. I would like to stop at traceability. Regarding the mentioned process, you said that some things are now happening outside the system, which complicates control and auditing. What auditing and tracking of request history options do you think the new system should offer to provide you with sufficient overview and transparency?

Respondent:
That's right, today we have a big problem with some communication or decision taking place outside the system - via email, phone, or even in person - and then no one records it anywhere. In the new system, the complete history of each request should be clearly available, including all statuses, time data and people involved. Every step in the process should be recorded - who approved, returned, edited or commented on what, when exactly it happened, and what was changed compared to the previous version.
Versioning is also important – the ability to view individual versions of a request if it has been revised. For example, if someone adjusts the amount, adds a new quote, or changes the

type of expense, a record of the previous status and the reason for the change should be maintained. This is also key for internal auditing and possible review by external authorities. Part of transparency should also include logging of system actions – for example, when an administrator changes approvers or adjusts a workflow – this too needs to be visible and traceable. Finally, we need to be able to easily export the request history as an audit trail – ideally in PDF or another clear format that we can attach to a contract or invoice, for example.
Such a level of traceability would greatly help us not only in resolving disputes or errors, but also in optimizing processes, because we would see where delays or errors most often occur.

Interviewer (Participant):
I understand, audit trail, versioning and logging of changes. Now I would like to jump to the perspective of an ordinary user. For example, a center manager who enters requirements, but is not necessarily an expert in the system. In your experience, what is the most complicated or confusing thing for ordinary users in the current system? Where do they most often get lost or make mistakes?

Respondent:
In our experience, the two most difficult things for ordinary users are orientation in what they actually have to enter, and how to correctly classify it from the perspective of budgetary and accounting logic. A typical center manager knows well what he needs to buy, but is confused about which type of expense to choose, cost center, or whether it is CAPEX or OPEX. If the system does not prompt or does not validate it, he either enters it incorrectly or asks instead - and this delays the entire process.
Another problem is that the current system does not guide the user step by step. The form is static and quite confusing – all the fields are there at once, regardless of the type of request. People often fill in things unnecessarily or skip something important because they are not clear about what is required. And when the system throws an error, it does not say exactly why – so the user has to guess.
Errors in the selection of approvers are also quite common, if the system leaves it partly up to the client. They either choose the wrong person, or they omit someone who is supposed to be there according to the rules. It would be ideal if the system knew these rules and built the approval path itself - based on the amount, type and center - and just showed it to the user.
And finally, we often hear that users don't know what's happening with their request at all - they lack an overview. They have to actively click somewhere to find the status, or they ask us. It would help if they could see a simple status right after logging in - "waiting for approval", "returned for revision", "approved, waiting to be ordered", etc. In simple terms, not in system terms.

Interviewer (Participant):
Great. I understand that the new system should guide the user step by step, dynamically display only relevant fields and validate the entered information. Let's stay with user-friendliness. If it were possible to design a so-called ideal entry form, what do you think it should look like from the perspective of an ordinary user? What should make their work easier, speed up entry and reduce the risk of error?

Respondent:
The ideal entry form should be as clear, contextual, and interactive as possible – that is, it should adapt to what the user is entering. Instead of seeing twenty fields at once, the system would first ask a few basic questions – such as “What do you want to purchase?” or “Is this

an investment, service, or material?” – and based on the answers, the form would display only the relevant fields. This would significantly reduce congestion and the risk that the user will miss something important.

The help would also be significantly improved by so-called "smart tips" - a small information bubble or contextual tooltip for each field that explains what to fill in, or gives an example. For example, for an ordinary user, the difference between a "cost center" and a "responsibility center" is often completely unclear.

The system should immediately suggest or pre-fill information it already knows – for example, the center according to the logged-in user, the budget line according to the selected purchase type, or the proposed approver. And if something is mandatory or entered incorrectly, it should be marked directly in the field, with a clear reason for the error.

The ability to save a proposal – i.e. fill out a request partially, save it and return to it later – and the ability to duplicate an older request as a template for a new one would also be a great help. This will save a lot of time for routine purchases.

And finally – the form should be optimized for common monitor resolutions and for mobile use, at least for basic actions. Not that anyone would enter a request from a mobile phone often, but for example, approval on the go or a quick check of the status of a request is a very practical thing for a manager.

Interviewer (Participant):

Great summary. Now I would like to return to approvers for a moment. We talked about their notifications and how sometimes clients choose the wrong people. In your opinion, should the approver have any decision-making support tools in the system? For example, summary information, risk alerts, direct access to the budget or history of similar requests? What could help him the most when approving?

Respondent:

Absolutely yes – approvers often make decisions under time pressure and without full context, so any supporting tool that makes it easier for them to navigate is a big plus. At a minimum, they should have summary information in one clear box at a glance – what is being purchased, for how much, for which center, from which budget, and what is the status of the item being used. Ideally, with a color indicator – green within the limit, orange just below, red above the limit.

Furthermore, the approver should have access to all attachments and the history of the request – who submitted it, who has already approved it, if it has been returned before and why. This often determines whether everything is OK or if it is necessary to take a step back.

It would also be useful to have quick access to a history of similar requests – for example, if it is a regular purchase, the system should offer previously approved requests with the same supplier or item type. This way, the approver can compare prices, terms, or frequency and evaluate whether the request is in order.

And if the request contains any potential risk – e.g. a missing attachment, an exceeded budget, an unclearly chosen type of expenditure or a missing link to a framework contract – the system should display a warning, ideally in the form of a so-called “risk flag”. This would prompt the approver to check the point in question without having to read the entire request in detail.

The goal is not to overwhelm the approver with information, but quite the opposite - to display the most important things clearly and in one place so that they can make decisions quickly and efficiently.

Interviewer (Participant):

Yes, a non-cluttered overview. Now I'd like to delve into the area of mobility and accessibility. You mentioned that mobile approval would be useful – do you think the new approval system should also be available as a mobile app, or would a well-adapted web version suffice? And for what types of tasks do you think mobile access would make the most sense?

Respondent:
In my opinion, a full-fledged mobile app would definitely be an advantage, but if the web version is responsive and fast, it would be enough for most cases. The important thing is that the user does not have to zoom in, click on miniature buttons or search for information on their phone - so the UI must be simple, clear and optimized for a smaller screen.
Mobile access makes the most sense for quick actions – typically approving a request, adding a comment, viewing a request detail or workflow status. Many managers or executives are often on the go, and the ability to "click" approval directly from their mobile phone will save them a lot of time and speed up the process for all of us.
On the other hand, in my opinion, entering a new request from a mobile phone is not that necessary - it is more advantageous to have it fully on the desktop, due to attachments, a more detailed description, or a more complex workflow.
But a "quick overview" function would be interesting – a notification from a mobile phone with the option to immediately open a simplified preview of the request with the main data and two buttons: "Approve" or "Return with comment". If the whole process takes less than a minute, it will increase the approver's willingness to respond without delay.

Interviewer (Participant):
Great, completely practical approach. Now I would like to move on to training and implementing the new system for a moment. If you imagine that a new system will be deployed in your organization, what do you think should be part of a successful deployment from the users' perspective? What would help them quickly understand the new system and adopt it in their daily practice?

Respondent:
In my opinion, the key to a successful deployment is well-targeted and practical training – not just a general presentation of the system, but specific scenarios according to roles. So different training for requesters, different for approvers and different for administrators. Each type of user should get a demonstration of exactly what they will do in the system and how – ideally in the form of short, interactive demonstrations or video tutorials, which they can return to later.
There should also be a simple user manual or wiki available – but not in the form of a long PDF, more like an online help with search and images. It would also be great to have contextual help directly in the system – so that when the user hovers over a field or button, they click on the question mark and a brief explanation or video is displayed.
It also helps a lot if there is someone in each department who has gone through the system in more detail – a so-called "superuser" or ambassador who others can turn to. It is not realistic that everyone will remember everything immediately after training.
And the first month after launch is absolutely crucial – there should be a dedicated support space, ideally a live helpdesk or chat, where users can quickly get an answer if they don't know what to do. If they encounter a problem in that first phase and no one helps them, they will lose trust and start looking for ways outside the system – which is exactly what we want to avoid.

Interviewer (Participant):
This is absolutely spot on, targeted training by roles, short and accessible tutorials. Now I would like to touch on security and permissions. How do you think user access to requirements should be handled in the new system? Who should have the right to see, edit, approve or export what? Do you have any specific requirements in terms of sensitivity or access hierarchy?

Respondent:
Yes, the issue of access rights is quite crucial for us – both because of sensitive data (e.g. supplier prices, investment plans, personal data in contracts), but also to maintain order in processes. Ideally, access should be controlled by a combination of the user's role, their placement in the organizational structure, and their current position in the approval workflow.
The requester should only be able to see and edit their own requests and their status, or requests from their center if they are the manager. They should only be able to edit the request until they submit it for approval - after that, they can only add comments, but not change the content.
The approver should have access to the requests that are currently waiting for their approval, but should not have the right to edit them - only approve, return or comment. After approval, they should only have read access to the requests that have passed their hands - for audit purposes.
Finance should have a broader approach – both to control requirements across the organization, but also for exports for reporting. We need to see not only the content, but also the history, workflow, budget impact and attachments. But finance should not be able to “bypass” the workflow either – that is, approve something that should not go through the standard path.
More sensitive requests – for example, for larger amounts, legal documents or contracts – should be accessible only to specific roles (e.g. legal department, directorate) and should be marked as “confidential”. The system should also support “need-to-know” logic – meaning that the user will only see what directly relates to their agenda.
And a very basic requirement: all actions must be logged, including unauthorized access attempts - precisely for the sake of an audit trail and internal controls. So access rights must be really strictly controlled and ideally connected to our personnel system, so that they are automatically updated when an employee changes position or leaves.

Interviewer (Participant):
Great. I'll stay with exports and data for a moment. What types of outputs would you expect from the system, whether for reporting, auditing or perhaps month-end closing? What formats, what data, how often and who should have access to them?

Respondent:
We would certainly expect the new system to provide us with high-quality and flexible outputs – both for operational needs, for monthly and quarterly closings and also for audit purposes. The basic requirement is the ability to export an overview of all requests for the selected period with key data – date of entry, amount, center, type of expenditure, budget item, approval status, who approved, time of approval, notes from the check and status of budget utilization.
We prefer Excel (or CSV) formats because we can continue to work with them internally - filter, group, prepare our own reports. PDF would be suitable for individual requests if we need to attach an approval record to an invoice or contract.

Furthermore, the system should be able to prepare a monthly summary for the financial statement - an overview of all approved but not yet implemented requests (e.g. orders that have not yet been invoiced), requests that are over budget or requests that have remained unanswered for more than X days. We need this as a basis for estimated items or internal reserves.

It would also be useful to have preset report templates - for example, "Requests by center", "Requests over 100 thousand", "Unclosed requests older than 30 days" - so that a regular user could pull them up without knowing export filters.

Access to this data should be limited to key people from finance, controlling and company management. Clients should only see their own requirements, and department managers should only see the requirements of their team. And of course, exports should be limited by permissions – for example, an HR user should not be able to export requirements from production.

And it would be ideal if some outputs could be planned – for example, a report would be automatically generated and sent by email to the responsible persons every month. This would save us routine work and ensure regular reporting.

Interviewer (Participant):
Perfect, very comprehensive. Now I would like to ask a slightly more strategic question at the end of this section. If you imagine that the new system is successfully deployed and works as you envisioned, what specific changes or benefits would you like to see clearly manifest six months after launch? What would you consider to be a sign that the implementation has been successful?

Respondent:
I would consider the implementation successful if several specific changes were visible within six months of launch, which would be visible not only to us in finance, but also to ordinary users. The first and most important thing is to speed up the approval process – if the average time from submission to final approval is reduced by 30-40%, it will be a clear signal that the system is working more efficiently and people are better able to navigate it.

The second sign of success will be a reduction in the number of errors or incomplete requests – that is, fewer cases where requests have to be returned due to misclassification, missing documents or budget discrepancies. This will mean that the system is validating, suggesting and guiding the user correctly.

The third thing we will look at is full system utilization – meaning that users stop bypassing the system with emails or manual solutions, and everything truly runs within one digital footprint. If we reduce the number of questions like “where is it hanging” or “who has to approve it”, that will be clear evidence that the new tool has improved transparency and trust in the process.

And last but not least, I expect that thanks to a better data structure and exports, we will be able to prepare documents for management or auditing faster - without manually pasting data from multiple systems. If the new system saves us dozens of hours per month on operations and control, it will be clear proof to me that we are on the right track.

Interviewer (Participant):
Thank you, that is a very concise summary of the benefits. Now I would like to move on to the last thematic area, which concerns a broader view of the digitalization of approval processes in your company. I will start simply. How do you think the company's management perceives the importance of digitalization of approvals and workflow management tools in general? Is it a strategic priority, or rather an operational necessity?

Respondent:
I think that the company's management is starting to see the digitalization of approval processes as a strategic priority - mainly because in recent years it has become clear how inefficient the current system is, and how much time, energy and nerves it takes up for the entire company. Previously, it was more of an operational necessity - that is, "something we have to have in order for it to work at all" - but today it is being discussed in the broader context of efficiency, transparency and risk management.
Management realizes that the approval system is not just a technical tool, but the backbone of corporate management – it affects costs, investments, cash flow and compliance. And when this backbone does not work well, it is reflected in all areas – from production delays to audit problems. So today, management understands that a quality workflow system can significantly contribute to better decision-making, faster response to business needs and greater accountability of individual departments.
At the same time, it is important to explain this strategy well to regular users – so that they do not perceive it as "another extra system", but as a tool that will make their work easier, not harder. And not only management plays a role in this, but also the way in which the entire project will be communicated and implemented.

Interviewer (Participant):
That's a very mature and balanced view, I completely agree with that. I'll stick to the strategic level. Do you personally have any vision or wishes for the future regarding approval and support systems in the company? What do you think the ideal digital ecosystem could look like in 2-3 years, if everything goes well?

Respondent:
Yes, definitely – I have a fairly clear idea in my head of where we could be in two to three years. In my opinion, the ideal situation is when today's "approval system" becomes a full-fledged platform for managing costs and decision-making processes across the company. So not just a tool where something is entered and someone approves it, but a place where budget, need, data and responsibility are connected in real time.
In the ideal ecosystem, all relevant systems – ERP, DMS, HR, purchasing portal – would be seamlessly connected, so the user works in one environment and does not have to switch between five different tools. They enter a request, the system offers them a supplier, checks the budget, alerts them to the framework agreement, ensures the correct approval path, and once everything is approved, it immediately creates an order, saves the documents and sends a notification to accounting. No manual rewriting, no delays.
At the same time, such a system should provide smart analytics - for example, overviews of the most common types of requests, bottlenecks in the workflow, prediction of budget utilization based on the current pace, etc. This is no longer just an operational tool, but a strategic tool for managing the company.
And what is important to me personally – I want the system to work naturally. So that users don't have to think about how to use it, but focus on what they want to do. This means maximum simplicity on the surface, but well-thought-out logic and connections inside. If we can achieve this, we will be much more efficient, faster and less frustrated.

Interviewer (Participant):
Really inspiring. You gave me a very deep insight into the current state and future needs. Is there anything else you would like to add or emphasize that we didn't mention during the interview?

Respondent:
Thank you, that's very good to hear - and I'm glad that the conversation wasn't just about "ticking off requirements", but that it went into depth. If I had to add something, it would be about one thing that is close to my heart and is often underestimated in projects: change communication.
Even the best system won't work if people don't own it. So I would like to emphasize that in addition to the technical solution, it is crucial to devote sufficient space to how we implement the new system - how we will talk about it, how we will work with concerns, how we will explain the benefits. Not everyone welcomes change, but if we give it meaning and the tools, most people will gladly accept it.
Otherwise, I think we've covered everything essential - if you have any additional questions at any time later, I'd be happy to come back to it. Thank you for the professionally conducted interview, it was also an interesting moment for me to reflect on what we actually want and need.

Conclusion of the interview with participant 5

Interviewer: Thanks again for the previous demonstration. I would like to follow up and ask you a little more in depth. I will start more generally – what development methodology do you primarily use?

Respondent: We operate more like agile, although not strictly according to Scrum or any other framework. Let's say it's a kind of pragmatic agility - we have regular standups, iterations after about two weeks, and retrospectives, but in a lighter form. It's definitely not a classic waterfall.

Interviewer: And what does this mean for you as an analyst in practice? How does this affect the way you collect requirements?

Respondent: It mainly means that we collect and refine requirements on an ongoing basis. We don't do a big analysis at the beginning of the project, but rather start with a skeleton and then gradually add to it based on priorities and feedback. The emphasis is on making the backlog a living document, not a dead list.

Interviewer: Looking at your interviewing style – what is your strategy when you want to get the right demands from people? How do you think about that?

Respondent: First, I try to understand the context – why the person needs something, not just what they say they want. I often do this by starting with very open-ended questions like “What is the biggest pain point for you in this process?” or “Where are you struggling the most today?” And then I go deeper with scenarios – “Can you describe the last time you dealt with this?” This often reveals real needs.

Interviewer: So you prefer specific examples?

Respondent: Exactly. People tend to talk in generalities, but specific examples are golden – that’s where you get the point across. And sometimes I’ll intentionally pause for a moment, or rephrase the question – that often leads to a better answer.

Interviewer: What types of requirements do you typically collect? Functional, non-functional, technical…?

Respondent: Most often functional – what the system should be able to do, what the user should see, what action should happen. But at the same time, in conversations we often come across non-functional ones – for example, requirements for speed, availability or security. Technical requirements are usually defined later, in cooperation with developers, but if something sounds like a technical limit already in the conversation, I make a note of it.

Interviewer: Can you give an example of when a non-functional requirement "split out" from a functional requirement?

Respondent: Sure. For example, if someone says, “I need approvers to see all of my team’s requests.” At first glance, it’s a functional requirement. But when I ask them why they want it, I find out that they want real-time visibility – so there’s also an accessibility and performance dimension to it. They want it to be online, fast, and maybe even accessible from a mobile device. That’s a non-functional layer.

Interviewer: Interesting. So when you're conducting an interview, do you have an outline or do you go more intuitively?

Respondent: I have a rough structure – a few main topics that I want to cover. But I adapt a lot to what comes up. I don’t want it to be like an interrogation. In fact, I usually prepare more questions than I can ask – but I’d rather sacrifice one or two than interrupt an interesting flow of thought.

Interviewer: What if you get a request that is very vague or unclear?

Respondent: In that case, I usually turn the conversation into a scenario – “Imagine it’s already working. What exactly would you do first? What should happen next?” This helps a lot to specify what they actually want. Or I use the “Given–When–Then” principle when I need to be more specific.

Interviewer: Great. To sum it up – agile approach, scenario-driven conversations, emphasis on concrete examples, layering of requirement types and working with the unspoken. Is there anything you would like to improve about your requirements gathering style?

Respondent: I would probably like to be even better at capturing non-functional requirements in the first interview. And sometimes I struggle to capture completely new needs that the user doesn't even think about. That's an art – being able to ask about things that haven't even occurred to the client. But I'm learning it all the time.

Participant 6

Beginning of the interview with participant 6

Interviewer: Hello, thank you for taking the time to talk to me. Before we begin – how are you today?

Respondent: Hello, thank you, I'm fine. A typical workday, but everything is going as it should. And you?

Interviewer: Thanks again, everything is fine. So let's get straight to it. How long have you been working as a business analyst?

Respondent: It will be exactly eight years now. It went by quickly, but I still enjoy it.

Interviewer: And how did you actually get this role?

Respondent: Well, at the very beginning I did something different – I started as a consultant, more of a generalist. But over time I started to focus more on analysis and requirements gathering, and then it just came naturally. I found that I enjoyed finding ways to make things work and helping to make them better.

Interviewer: So you had another position before that?

Respondent: Yes, as I said, originally a consultant. But I found my way into analysis.

Interviewer: What types of projects do you have the most experience in? IT, process analysis, strategy...?

Respondent: Probably mostly IT – mostly it was about implementing various systems or modifying internal tools. But I also worked on projects with overlaps into process analysis, for example when changing company workflows. Strategy only marginally.

Interviewer: What about your work experience – did you work in one company, or did you change environments more often?

Respondent: I changed jobs a lot. I was self-employed, so I worked for more than ten companies. Most of them were large corporations, over 700 people. So I had the opportunity to see different approaches, which I consider an advantage.

Interviewer: And are you currently in a large or small company?

Respondent: Currently, it's a large company – around a thousand employees. I prefer this environment because there are set processes and at the same time room for improvement.

Interviewer: Thanks, so we can get started on the experiment we've prepared. Here's your assignment, and once you're ready, we can begin.

Participant-Respondent Interview (ChatGPT)

(https://chatgpt.com/share/67eaea40-0f84-8008-b312-352dbf78b5c4)
Interviewer (Participant):
Hello, thank you for taking the time to talk to me. First of all, I would just like to ask you how you actually feel about the current approval system? What do you find most annoying or annoying about it?

Respondent:

Hello, thank you for your attention. The current system has honestly not met our needs for a long time. What bothers me the most is that it is slow and confusing – approval often gets stuck because it is not clear where exactly the document is “hanging”. If someone forgets to approve, they do not receive any notification, so we have to do it manually, which is a waste of time. In addition, there is no change history or clear logging of steps, so tracing back the reasons for approval or rejection is complicated, especially when it comes to older things. And I am not even talking about the fact that the system sometimes “crashes” itself and then we have to ask IT to restart it.

Interviewer (Participant):
Thanks for your honesty, this is very valuable. And when you say it's confusing. What do you mean specifically? Is it more about how things are displayed, or are there some filters, dashboards missing, or is it hard to find information?

Respondent:
Yes, it's a combination of everything, but mostly it's just hard to navigate. The user interface is outdated and illogical - you have to click through several windows before you get to the details of a document. There are no basic filters by document type, date, approval status or department, so when I'm looking for a budget proposal from last quarter, for example, I have to go through dozens of records manually. There are also no dashboards or a clear queue of "what's waiting for me" versus "what I've already dealt with". There is practically no summary or analytical view of how many approvals are in progress, where they are most often delayed, or how long individual steps take. I would like to see all of this in the new system - ideally right after logging in.

Interviewer (Participant):
That makes sense, thanks. And as for the approval process itself. Can you describe to me what the life cycle of a document typically looks like? Who submits it, who approves it, how many steps are there, and who all participates in it?

Respondent:
Sure, I'll take the example of a purchase request, because we deal with it often. The assignment is usually carried out by an assistant or clerk from individual departments - for example, IT, marketing or operations. They fill out the form, attach attachments, such as offers from suppliers, and send it to the system. The first approval step goes to the head of the department, who checks the legitimacy of the request. If it passes, the document goes to the finance department - there it is approved by either me or one of my colleagues, depending on the amount and budget category. For higher amounts, it is then approved by management - usually by the financial director or executive director. There are usually 3 to 4 steps in total, but the problem is that the current system cannot flexibly set up different approval workflows depending on the type of request or the amount - everything runs according to one template. So sometimes we unnecessarily burden management with small things, and other times the system does not force the document to go through all the necessary levels.

Interviewer (Participant):
Thanks. And when you say that the system cannot set up different workflows. What kind of flexibility would you ideally like? For example, having the ability to set it up by document type, or having the amounts automatically determine who should approve it?

Respondent:

Exactly. It would be ideal if the new system allowed us to set up different workflows based on the type of document – a different procedure for a purchase request, a different procedure for a budget change, a different procedure for an investment plan. And at the same time, the process could be branched based on the amount or even the type of purchase – for example, for operating expenses up to a certain amount, it could end at the level of the department head, but for larger amounts, it would automatically go through finance and management. It would also be great to be able to define exceptions or add ad hoc approvers, such as a lawyer or technical expert, when a contract or technical documentation needs to be reviewed. I simply want the system to be able to mimic how our decision-making naturally works – with a certain logic, but also with the possibility of intervention when the situation requires something special.

Interviewer (Participant):
I understand. And regarding those ad hoc approvers. How do you think their involvement should ideally work? Should someone add them manually during the process, or should the system itself suggest that hey, here we need a lawyer based on the type of document or attachment?

Respondent:
Ideally, a combination of both. In some cases, the system should automatically suggest involving someone else – for example, if there is a contract in the attachment, it should recognize it either by the file type or keywords and recommend involving the legal department. The same goes for technical tasks, where a technician or IT person should step in. But at the same time, I want to be able to add someone manually – for example, when we know that the opinion of a specific person who does not normally intervene in the process is needed. But it has to be simple, not that I will search in some complicated list – ideally by name or role. And I would also like it to be clear whether the ad hoc person has the right to approve or just add a note, because that is the difference – a lawyer is often only supposed to comment, but the final approval remains with management.

Interviewer (Participant):
The difference between commenting and approving is important. Speaking of which, how do you actually handle it now? Do you have notes or emails with comments somewhere, or is it all done outside the system?

Respondent:
Unfortunately, this usually happens outside the system. When we need a lawyer or someone from the technical department, we send documents by email or via Teams and then copy their comments to the side, or we manually rewrite them into the system as a note. Of course, this is not ideal - on the one hand, the context is lost, because it is not clear who commented on what, and on the other hand, it is not visible in the system at all that someone else entered there. Moreover, it is quite dependent on whether a person forgets to add it, so from the point of view of auditing or retrospective control it is inconclusive. In the new system, I would like to have the option to add commenting roles, whose inputs are logged, but at the same time to make it clear that they are not approvers - so that the process is not delayed, but at the same time that the comment has weight and is visible to everyone else in the chain.

Interviewer (Participant):

Sure, to make it transparent. And when you talk about back-checking. How often do you need to track who approved what, when it was, or why it was rejected? And what do you typically need it for?

Respondent:
We look for this quite often, mainly for audit reasons, internal controls or when discrepancies appear in billing. For example, we find out that a service was ordered that was not approved according to the correct procedure, or that the budget was exceeded and we need to find out who gave the approval. In such moments, it is crucial to have an overview of who approved or rejected the document when, and ideally with a justification. Now it is a problem because the system shows who was in the chain, but often there is no exact time stamp and it is not mandatory to add a comment when rejecting or returning a document. So we do not know in retrospect what went wrong or why the decision changed. In the new system, I would definitely like to have a detailed log of every step - who, when, what was done, including the reason if the document was returned or rejected. In my opinion, this is absolutely fundamental.

Interviewer (Participant):
I understand, without proper logging, you're just fumbling around. And as for comments. Do you think they should be mandatory, for example when rejecting or returning a document? Or should it depend on the type of action?

Respondent:
For me, this should be mandatory at least when rejecting and ideally also when returning a document for completion. In these cases, it has a fundamental impact on the further procedure - the client must know what to correct or why it did not pass, and at the same time others in the approval chain should see what the reason was. For regular approval, a comment may not be mandatory, but it should be possible - for example, if the approver wants to add a note like "I approve, but only up to the amount of XY" or "provided that the contract is amended according to the lawyer". But for rejection and return, I would really enforce it - without a comment, it loses its meaning.

Interviewer (Participant):
I agree, without an explanation it's a complete mess for the client. Now a bit from a different perspective. What about notifications? You said at the beginning that it's a problem when someone forgets to approve something. What types of notifications would you like to have in the new system? And in what way – email, notifications in the system, mobile applications...?

Respondent:
Notifications are key for us, because their absence is often the reason why approvals are unnecessarily delayed. Ideally, I would like a combination of system notifications directly in the application, but also emails, because not everyone in the team sits in the system all day. A mobile application with push notifications would also be useful for key people, especially for management who often travels or is in meetings. And as for the content of the notifications, I want them to be meaningful. Not just "a document is waiting for you", but to contain basic information: the type of document, who it is from, the amount, the status, and perhaps even the deadline, if approval is time-critical. And it should also be possible to set reminders - for example, if a document is not approved within 2 days, an automatic reminder will be sent. Ideally, escalation could also be set - if someone ignores it for a long time, the system will send it to their superior.

Interviewer (Participant):
Yeah, sure. Speaking of mobile apps, are there any tools you currently use to communicate or handle approvals outside of that system? Like Outlook, Teams, Slack? And should the new system somehow connect to these platforms?

Respondent:
Yes, we use Outlook and Teams most often. We send comments, additional materials or reminders by email when the system does not notify us on its own, and we often informally discuss who approved what or why something was delayed through Teams. It would be great if the new system could connect to these tools - for example, so that notifications could also come to Teams, ideally with the option to view details or approve directly from the notification, at least for simpler things. And it would be nice to have automatically generated emails in Outlook that would not just be a simple link, but would contain key information, as I said - amount, document type, deadline, etc. If the system also had an Outlook plugin or some way to quickly pull approvals into an email without switching windows, it would be a big plus. Connecting to these tools would significantly simplify our daily work.

Interviewer (Participant):
This connection certainly makes sense. And now I'll ask a little about the user side. What do you think is the most important thing for people to quickly get used to the new system and not resist using it? What should it have so that they take it for granted?

Respondent:
The most important thing is that the new system is intuitive and fast – the user must immediately understand what to do, what the system wants from them and where to find what they need. If it is complicated or slow, people will find ways to get around it. The interface must be clear, ideally with a dashboard where everyone can immediately see what they have to do, what they have sent and where it is. The second thing is that the system makes work easier – not just another layer of bureaucracy. If it offers templates, pre-filled fields, automatic data filling (for example from the budget or previous approvals), people will like it. And of course, fast response, stability and the ability to use it on mobile also play a role. If all this works, the transition will be much smoother.

Interviewer (Participant):
Great. And I'm also curious. Do you have someone on your team or in your company who is having really big problems with the current system? A typical power user or, conversely, someone who often complains or uses it completely differently than others?

Respondent:
Yes, we have a few people in the team who are at opposite ends of the spectrum. On the one hand, there is a colleague from controlling who is a so-called power user – she uses the system daily, has an overview of it, but at the same time is frustrated by how many things she has to do manually. She often gives us suggestions like “we could automatically add a budget item here” or “there should be a validator here so that it cannot be sent incompletely”. On the other hand, we have managers from some departments who approve once in a while and find the system complex and unintuitive – they often call or write to us that they don’t know where to find what, or why something can’t be sent to them. And then there is the typical case when someone bypasses the entire process – they send the approval by email, and we then rewrite it into the system, just so that something “is there”. So it certainly makes sense to

involve both the experienced users and the “resisters” when setting up the new system – everyone has a different perspective and both are valuable.

Interviewer (Participant):
That sounds like the perfect combo for a pilot. One who knows every nook and cranny of the system, and one who sees it through the eyes of a regular user. And then there's the templates and autofill you mentioned. What type of information do you think the system should automatically prefill? Where should it get it from?

Respondent:
Exactly, templates and pre-filling are key to speeding up work for me. Ideally, the system should automatically fill in information based on who is entering the document and what type of request they are filling in. For example, if an IT officer is filling in a purchase request, the system should pre-fill in the cost center, budget item, expense type, and perhaps even the approval workflow – all of which can be taken either from the user profile or from the history of previous requests. It should also be able to automatically fill in document number series, the date of entry, and ideally load the current budget status of the given department – so that the user can see how much is left in the budget before sending the request. In the case of a recurring expense, it would be great to offer the option to “create a copy” of a previously approved document and just edit the amount or description. In short, handle as many routine things as possible automatically and let the user deal with only what is really new or specific.

Interviewer (Participant):
Okay. And speaking of history and repetition. Would you like the system to be able to provide some reports or statistics? Like how many approvals were made per month, how long did each step take, who has the longest delay, and so on?

Respondent:
Certainly yes, I really miss this in the current system. I would like to be able to generate reports such as: how many documents were approved in a certain period, how many were rejected, how many returned funds for addition, how long each approval round took - the average and specific cases are fine. Statistics by person or role are especially important - who approves on time, who often delays, where things "break down" most often. I would also be interested in how many documents are processed without intervention (e.g. by automatic flow) and how many require ad hoc solutions. Ideally, the system should have preset basic reports, but also the ability to set your own filters and export data to Excel or Power BI. These outputs are important not only for department management, but also for improving processes and advocating changes to management.

Interviewer (Participant):
Sure, I understand. Now a little differently. Do you ever find that people approve things just to get it off their chest, without really going through it? And if so, how do you think this could be prevented?

Respondent:
Yes, unfortunately it happens, especially when people are very busy or when the system does not allow easy access to the details of the document – then they simply click “approve” without going through it thoroughly. Sometimes they also rely on the fact that “someone has already checked it before them”, which is quite dangerous. This could be prevented in several ways: first, having a clear preview of the document directly in the notification or on the main

page of the system, without having to click through x windows. Second, it would help to introduce a certain “checklist” or mandatory fields before approval – for example, confirming that the approver has read the attachments or checked the amount. And third, it could also work if the system highlights changes compared to the previous version of the document if it is a repeated approval. Simply motivate people to know what they are approving – not just “unmark it”.

Interviewer (Participant):
Yeah, I understand, so that it's not just a click and go, but a really conscious decision. And about the attachments. How do you work with them now in the current system? Can they be easily uploaded, displayed, and edited? Or is it a struggle?

Respondent:
Attachments are a real struggle right now. Uploading works, but it's quite clumsy - the system can't handle multiple files at once, doesn't support larger formats, and sometimes the file doesn't load properly, but the system doesn't notify you of this. But what's a bigger problem is working with attachments after they've been uploaded - they can't be properly displayed directly in the system, you have to download them, which is inconvenient and time-consuming, especially on a mobile phone or outside the office. There's also no version control, so when someone uploads a newer version of a contract or budget, the older one simply remains there without a label, and it's not clear which one is valid. In the new system, I'd like to have a preview of attachments directly in the workflow, the ability to overwrite versions or keep them with changes marked, and ideally, comments on individual attachments. It would also be nice if the system could recognize duplicate attachments or warn you if a mandatory document is missing - for example, a contract for higher amounts.

Interviewer (Participant):
Yeah, that sounds like a lot of room for improvement. And now I'm curious. Do you need a digital signature in some of your processes? And if so, how would you imagine that working in the new system?

Respondent:
Yes, in some cases we really need a digital signature – most often for contracts, budgets or documents that go outside the company, for example towards suppliers or partners. So far, we have been dealing with this in a rather unsystematic way – the document is printed, signed by hand, scanned and uploaded again, which is inefficient and time-consuming. Ideally, the new system should have integrated support for digital signatures – either through our existing platform (e.g. Signer, DocuSign or a qualified certificate), or the ability to connect to a state-recognized service. The process should be simple: the document is prepared, approved, and when it is to go out or a signature is needed, the user will receive a prompt for a digital signature – ideally with multi-level authentication. It is important that it is clear who signed what, when, and that it is legally admissible, perhaps even in the sense of eIDAS. And also that the signature “locks” the given version of the document so that it cannot be edited after signing.

Interviewer (Participant):
Perfect, this is concrete and technically feasible. We are slowly approaching the end, so something about integrations. Apart from those with Outlook and Teams, is there any other system that the new approval system should connect to? For example, ERP, accounting, document management, something like that?

Respondent:
Absolutely yes – connection to ERP is absolutely key for us. We use Helios, and ideally I would like the approval system to be able to both draw data (e.g. current budget, code lists, centers) and send data back – for example, approved requests or budget changes, which are then reflected in accounting. This would eliminate unnecessary double entry. There should also be a connection to our internal document management system (DMS), where we archive contracts and other documents – ideally so that approved documents are automatically saved in the right place with the version, metadata and link to other related files. And if we were to implement a system for electronic public procurement, it would be nice to have the option of integration here too – so that the requests or documents can be seamlessly transferred to the tender process. The more connections, the less manual work and the greater the consistency of data.

Interviewer (Participant):
This is a great approach. One last question to conclude. If you had to say one thing that really had to change in the new system compared to the old one, what would it be? Simply the most fundamental thing, without which it wouldn't make sense to you.

Respondent:
Clearly, transparency and traceability of the entire approval process – I want to know exactly who approved or rejected what, when, with what comments, and where the document is currently located. Without this, responsibility is lost, confusion arises, and the process cannot be effectively managed or controlled. Even if everything else were better, but this was missing, it would not be a step forward for me.

Interviewer (Participant):
Perfect, thank you very much for your honesty and all the detailed answers. I have a lot of specific suggestions from you about what is missing in the system and what the new one definitely needs to be able to do. If you have anything else that comes to mind in retrospect, let me know.

Respondent:
Great, thank you too – I'm glad you're taking it so systematically. If I think of anything else, I'll definitely get back to you, or we can meet up to discuss some prototypes or a sample solution, I'd be happy to give feedback in the next phase. Good luck with the design!

Conclusion of the interview with participant 6

Interviewer: Thanks for the demonstration, it was really interesting to see how you do it in practice. Now I would like to delve a little more into your processes. What development methodology do you actually use?

Respondent: I work exclusively in an agile environment. Mostly within SCRUM, sometimes Kanban. It depends on the team, but always agile.

Interviewer: And how does that affect the way you conduct interviews when collecting requirements?

Respondent: Agile is very much about iteration and collaboration. So I conduct interviews more like a conversation than an interrogation. I don't try to get everything right away, I just figure out the key needs, and we add the rest gradually in backlog refinement or during the sprint.

Interviewer: Do you have a strategy for asking questions in a way that gets the right requests out of people?

Respondent: Absolutely. I use a lot of open-ended questions, like “How would it ideally work…”, “What didn’t work for you in the current solution?” And I often work with examples – I simulate scenarios, I give hypothetical situations. I also ask more “why” than “what do you want”. I want to understand the motivation, not just the output.

Interviewer: And do you ask questions in a more structured way, or do you let the conversation flow freely?

Respondent: It depends on the context. If I'm talking to someone in business, I often start loosely to get to the heart of the problem. Once we have a framework or backlog, I go more structured by areas. But I always want it to feel natural.

Interviewer: What types of requests do you most often try to get during those interviews?

Respondent: Primarily functional requirements – what the system should be able to do. But at the same time, during the interview, I also map out non-functional requirements – for example, requirements for availability, usability or security. I usually coordinate technical requirements with the developers, but if they result from limitations of the infrastructure or existing systems, I take them into account from the beginning.

Interviewer: Do you have a system for sorting or categorizing these requests?

Respondent: Yeah, I usually sort them into three layers – business requirements, user requirements and system requirements. I also mark them by priority – either according to the MoSCoW method or using WSJF if we are in a SAFe environment.

Interviewer: What do you do when the client is unsure or can't say exactly what they want?

Respondent: This is a common case. I help him visualize the situation – either with a visualization, a prototype, or I guide him through a story: “Imagine you are a user and…” Sometimes I also use a checklist of typical areas so that I don’t miss anything.

Interviewer: And how do you resolve conflicts between the demands of different stakeholders?

Respondent: I try to get them to sit at the same table, or at least confront them with the other side's perspective. This often helps to realize the impacts. If it can't be reconciled, I write down both options and recommend a compromise - sometimes the MVP approach helps, other times the PO or management decides.

Interviewer: How do you document requirements?

Respondent: In agile teams, mainly in the form of user stories. If something more detailed is needed, I supplement the acceptance criteria or create a specification alongside. We use Confluence and Jira, so everything is traceable.

Interviewer: Do you prefer individual interviews or group workshops?

Respondent: Both have their place. Individual ones are good for a deeper understanding of a role or specific needs. I use workshops when I need to align multiple parties or when we are fine-tuning the flow across departments. The advantage of workshops is speed – disagreements are quickly revealed.

Interviewer: And lastly – do you record the interviews, or do you just take notes?

Respondent: If possible, I record interviews – with consent of course. It's much more accurate than just taking notes. But for quick meetings, I just take a brief note. I then transcribe the important ideas straight to the requirements.

Participant 7

Beginning of the interview with participant 7

Interviewer: Hello, thank you for taking the time to talk to me. Before we start with the main questions – how are you today?

Respondent: Hello, thank you for asking, I'm fine. A bit of a busy day, but nothing that couldn't be managed. And you?

Interviewer: Thanks again, everything is fine. So let's start - how long have you been working as a business analyst?

Respondent: It's been about four years now that I've been fully dedicated to it.

Interviewer: And how did you actually get the role?

Respondent: Well, it wasn't exactly planned. I originally joined a company as a junior project coordinator, but gradually I became more involved in analytics and user communication. Eventually I was invited to design a new internal system, and since then I've moved more towards an analytical role.

Interviewer: So you had a different position before?

Respondent: Yes, as I mentioned, I started out more in project management, but even then I enjoyed finding out what people actually needed and how it should work more than the planning and management itself.

Interviewer: And in what types of projects do you have the most experience? For example, IT, process analysis or strategy?

Respondent: IT definitely predominates. I worked a lot on system implementations, integrations and process automation. But at the same time, there was also a fairly strong component of process analysis – mapping, optimization, introduction of new workflows. I touched on strategy rather marginally.

Interviewer: Have you worked in different companies, or do you have long-term experience in one organization?

Respondent: I've been through six companies so far. Most of them were project or contractor positions, so I naturally moved up based on what was needed. But it was a good experience, you see different environments, different approaches.

Interviewer: And those companies – were they smaller or larger?

Respondent: They were always larger companies, mostly corporations with around six hundred employees or more. So the environment was often quite structured, with clearly defined roles and processes.

Interviewer: Thank you very much for your answers, so we can start the experiment we have prepared. Here is your assignment and once you are ready, we can begin.

Participant-Respondent Interview (ChatGPT)

(https://chatgpt.com/share/67ec1bee-2d80-8008-bffd-fbe383bfd259)
Interviewer (Participant):
Hello, thank you for taking the time. To start, could you please describe how you currently use the approval system in your daily work? What do you like most about it?

Respondent:
Hello, of course. We use the approval system primarily to approve budgets, purchase requests, contracts, and internal financial documents, such as transfers between centers or planned investments. In practice, it works like this: the applicant fills out a form, attaches the necessary documents, and the system automatically determines the approval workflow based on the type of request and organizational structure. What I appreciate about the system is that we have at least some level of automation and traceability - we know who approved what, when, and what status the request is in. I also appreciate that we have the ability to enter comments and that the system sends notifications by email, although it is not completely reliable. But now we are getting to what does not work ideally on it. Do you want to hear it right now?

Interviewer (Participant):
Thank you for the detailed description. I would definitely like to hear what is not working ideally in the current system. So what specifically would you improve?

Respondent:
Certainly. There are more, but I'll start with the most important one - the system is slow and technically outdated. It has compatibility issues with modern browsers, it often crashes or freezes, and sometimes the data doesn't save correctly. This causes not only frustration, but also delays, as users have to re-enter requests or contact IT support. From the finance department's perspective, it's also a problem that we can't see related budget data at a glance - for example, how much has already been spent in a given chapter, whether we have a reserve, or whether the request exceeds the plan. We lack a clear control screen with the ability to

quickly filter by center, type of expense, or approval status. And I would also appreciate better connectivity to the accounting and ERP system - today we have to search for some data manually, which is inefficient and prone to errors. Should I continue with other restrictions?

Interviewer (Participant):
Thank you. Please continue, I'm curious what else you would improve or what is giving you trouble.

Respondent:
Another big problem is the lack of flexibility in approval workflows. The system has fixed approval routes that cannot be easily adjusted to a specific situation. For example, if someone is on vacation or sick, we have to deal with it through the IT department or formally outside the system, which disrupts the entire process. There is no possibility of delegation or temporary replacement directly in the interface. It is also not possible to set conditions based on the amount or type of expense - for example, small purchases up to a certain amount should go through a shortened approval process, while larger investments should have a multi-level process.
Another thing that complicates our work is the lack of change records. The system records who approved what, but not always who edited or commented on what, so tracing back is not reliable. In addition, I would like the option to return a request with a specific comment, instead of just rejecting it, so that the requester knows exactly what to edit and can easily continue. And then there is the user-friendliness - the interface is outdated, unclear, and we have to train new colleagues because it is not intuitive.

Interviewer (Participant):
That's a lot of important points, thank you. Focusing on user-friendliness, what specifically do you find most confusing or difficult to understand about the interface for new users? What would you imagine as the ideal user experience?

Respondent:
The most confusing thing is that the user often does not know what to do or where to find something. Forms are not structured - fields are not logically ordered, some are mandatory, but it is not immediately visible, and error messages are often unclear or appear only after submission. It is also not clear what stage the request is currently in - the workflow is "hidden" and there is no visualization of the progress or a simple timeline. New colleagues are often not sure whether their request has really been submitted or is waiting for some further action from them.
For me, the ideal user experience would be based on a simple, intuitive interface – ideally with a wizard that helps step by step. The form should be logically organized, with context-sensitive help, real-time validation and the ability to automatically pre-fill some fields (e.g. organizational units, centers, code lists). The user should see at a glance what to do, what the status of their request is and what happens next. And as a manager, I would like to be able to easily filter requests, receive overview dashboards and have access to history and statistics without having to export to Excel.

Interviewer (Participant):
That's a great overview, thanks. If we focus on those dashboards and reports, what exactly would you like to have available as a finance manager? What specific metrics or information would you need to have at your fingertips to have better control over your processes?

Respondent:
I would definitely like to have an overview of the total volume of approved and unapproved requests, divided by type (e.g. purchases, investments, operating expenses), center and approval status. I would also like to know how many requests are in which stage of the process – how many are waiting for approval, where they are, who is currently "on the move". Ideally with the ability to drill down to specific items. Metrics such as average and median approval time, identification of bottlenecks or repeatedly delaying approvers are also important to me.
From a budgetary perspective, I need to see how much has been spent against the plan in individual budget chapters, in real time. I would like to be able to compare approved requests with the budget limit and monitor whether the center is not spending beyond the limits. Exporting to PDF or Excel for reporting would also be useful, or the ability to set up automatic sending of monthly reports to e-mail. And ideally, I would have the ability to customize the dashboard - for example, in the form of widgets depending on what is important for me to monitor.

Interviewer (Participant):
Thank you. What specific problems or frustrations do you have when using the system, for example in terms of collaboration between departments? For example, in communication between you and other team members, such as IT or purchasing?

Respondent:
From the perspective of collaboration between departments, one of the main problems is that the system does not offer any space for effective communication directly within the request. Comments are there, but they are not linked to specific actions or people, there is no notification of a new message, and we often do not even notice that someone has added something. So it happens that we have to deal with it by email or verbally, which loses the connection to a specific request and complicates tracing the history. It is also not possible to tag a specific person in a comment, so no one feels responsible for responding.
At the same time, there is a lack of transparency in what is happening with the request in another department - for example, if we approve the budget framework, but the purchasing department is still negotiating with the supplier, in the old system we have no overview of where it is "hanging" or whether anything has been done. The problem with the IT department is that every workflow adjustment, new field or change in the interface requires the intervention of a developer. This holds us back because we cannot respond flexibly to changes in processes. It would be ideal if each department had its own clearly defined space with shared status and the possibility of collaboration, but without the need to solve everything outside the system.

Interviewer (Participant):
Thank you for elaborating on these issues. When you talk about this shared collaboration space, what would be the ideal solution for you? Should it be some specific place for each department where they can track status, add comments, and easily inform each other? What would be your ideal process?

Respondent:
Exactly – the ideal solution for me would be for each requirement to have its own "work card" or detail page where everything would be in one place. Each department involved in the approval or processing would have access to their part – for example, the purchasing team could add the status of the request or supplier selection, IT could note the status of the

implementation or configuration, and the finance department would have an overview of the budget allocation. Everyone would see the status of the other steps at the same time, but with clearly defined responsibilities for their parts.
Comments should be structured – for example, by stage or responsibility – and it should be possible to tag a specific user to receive a notification. It should also be possible to attach attachments or links directly to a given stage or activity so that they don't get lost. And I would like every action in the system to have an audit trail – who changed what, when and why. My ideal process is one where everyone sees what to do, when to do it, where it is located, and where everything important stays with the specific request – not on the side in an email or in your head. This will significantly increase clarity, accountability and efficiency.

Interviewer (Participant):
This sounds like a great solution for ensuring clarity and efficiency. How would you imagine the system responding in case of any problems or failures? What form of support or notifications would be most convenient for you so that you can quickly find out what happened and how to fix it?

Respondent:
Ideally, the system should be able to proactively inform about a problem – not when we notice it, but as soon as it occurs. For example, if the approval workflow encounters an error (e.g. missing approver, incorrectly entered data, system unavailability), I should receive an immediate notification with a clear description of the problem, ideally by email and within the system itself. I would also like a simple dashboard or overview of "problem requests", where all blocked or error states would be visible in one place.
From a support perspective, I would appreciate it if the system offered context-sensitive help – e.g. why a given step cannot be performed, what needs to be fixed and what the recommended next step is. If it is a technical error, the administrator should be able to easily find out what happened, and the user should be able to report the incident directly from the interface, without having to write long emails to IT. Ideally, the system would also offer a simple helpdesk module or a link to internal support with pre-filled information about the request, to reduce the resolution time. Overall, I would like the system to not only support normal operation, but also handle exceptions and errors well – because these are the ones that delay the most.

Interviewer (Participant):
That sounds interesting. What are your thoughts on user training for the new system? What specific training or support would you consider necessary for users to easily adapt and start using the system effectively?

Respondent:
I know from experience that the success of an implementation often depends on how quickly and clearly users learn to use the system. That is why I consider it essential that training takes place in multiple forms and is tailored to different groups – training will look different for regular users, different for managers, and completely different for administrators or superusers. I would definitely like to have a clear manual or online help with clear screenshots and step-by-step procedures, ideally an interactive guide directly in the system.
I would also appreciate short videos or microlearning modules that people can watch on their own as needed. Practical training should ideally be done in person or as a webinar, where users can try out the system right away. It is also key to have someone in the beginning who will be the so-called "first aid" - that is, someone from our department or IT, who will be

available for quick questions in real time, at least for the first month after launch. And I would definitely appreciate the possibility of feedback on the training and the system itself - so that we can fine-tune what does not work optimally right away.

Interviewer (Participant):
Thanks, that makes sense. What are your experiences with testing a new system? What should the new system allow for in terms of testing to ensure that everything works as expected before it is fully implemented?

Respondent:
I have experience with testing and I know that this is a phase that should not be underestimated - otherwise errors or shortcomings will only become apparent in live operation, when it is much more expensive. For me, it is crucial that the new system allows for thorough user testing in a realistic environment, ideally with real data or an anonymized copy of it. We must be able to test complete approval processes, various scenarios - including exceptions, escalations, rejections, delegations, etc. It is not enough to test only the "ideal path", we also need to verify the behavior of the system in the event of errors or non-compliance with conditions.
Testing should be enabled across roles – not just from the perspective of a regular user, but also from the perspective of an approver, administrator, and reporting employee. The system should allow for easy feedback entry – ideally right in the testing interface, so that developers and the project team have specific suggestions directly linked to a specific screen or function. And it is very important that the time for testing is sufficient and not just driven by the launch date – it is better to postpone live operation by a few weeks than to risk chaos and loss of user trust right from the start.

Interviewer (Participant):
That's a great approach. What methods of communication or coordination between teams would you consider necessary during the implementation to ensure the whole process runs smoothly and without unnecessary complications?

Respondent:
I consider it crucial to have a clearly defined project management structure – i.e. specific people responsible for individual areas (finance, IT, purchasing, HR, etc.) who will act as main points of contact. These people should be in regular contact – ideally in the form of short weekly status meetings, where progress, obstacles and tasks for the next period are addressed. Communication should be centralized – e.g. using a shared platform such as Teams, Confluence or a project tool such as Jira or Asana, where there will be an overview of all tasks, versions, notes and comments.
At the same time, it is important that communication is not only between the implementation team, but also towards regular users - i.e. regular information about what is coming, what impact it will have, when there will be training, when testing, when the transition to live operation. Users must feel that they are part of the process, that they can influence something and that their comments are meaningful. It is a good idea to create a channel for collecting questions and feedback - whether in the form of a form, chat or live Q&A. And last but not least, the availability of key people must be ensured during critical phases - especially during testing and live launch, so that problems can be quickly responded to.

Interviewer (Participant):

Thanks for the great answer. What specific features or characteristics should the new system have to meet your team's needs and be well-suited for developers? What should the new system be able to do in terms of ease of maintenance, scalability, or the ability to adapt to specific requirements in the future?

Respondent:
Thank you, that's an important question. From our team's perspective, the new system should definitely support configuration without the need to intervene in the code - i.e. the so-called no-code or low-code approach. Ideally, the administrator should be able to modify the workflow, add new fields, change approval rules or set notifications without the need to involve developers. This gives us flexibility when changing processes, which in practice come up quite often.
In terms of scalability, the system should be ready to handle an increase in the number of users and requirements without impacting performance. It should also allow the addition of new document types or extensions to other departments without the need for a complete redesign. Modularity is very important - the ability to turn on and off functionalities as needed, so that the system is not overwhelmed with unnecessary features that no one uses.
For developers, it is crucial that the system has a well-documented API – for integrations with ERP, accounting, DMS or SSO. We want the system to communicate with other tools we already use, and ideally support webhooks or other modern integration standards. From a maintenance perspective, transparent version management, the ability to test new features in a sandbox, and clear logs that will allow you to quickly find errors are important. And of course, robust roles and permissions to fine-tune who sees what and what they can do – this is absolutely essential for finances and security.

Interviewer (Participant):
Thank you. Focusing on specific security requirements, what would you consider to be the most important features or measures that a new system should have to be secure enough for your needs?

Respondent:
From a security perspective, it is absolutely essential for us that the system supports single sign-on (SSO) linked to our corporate identity - that is, that users log in using their corporate accounts and that their access can be centrally managed. We also require support for two-factor authentication, at least for administrative roles or access to sensitive information.
Detailed permission management is very important - i.e. the ability to set exactly who can see which requests, approve, edit or download documents. The system should have an audit trail of all operations - who created, edited, approved or rejected what, including changes to settings. We also need data to be stored encrypted and transmitted using a secure protocol (e.g. HTTPS, TLS), and to be able to set a data retention policy according to our internal rules and legislation.
From a compliance perspective, the system must meet GDPR requirements – i.e. the ability to track who has access to personal data, and the ability to anonymize or delete data if necessary. We would also welcome the possibility of regular security audits or penetration tests, and it is also important that the system supplier guarantees high availability and regular backups, including the ability to quickly restore data in the event of an outage.

Interviewer (Participant):
Thank you for the detailed overview of security requirements. As we focus on the implementation of the system, what would you consider to be the key steps to ensure that the

transition is as smooth as possible and has the least disruption to the normal operations of the business? What specific phases would you recommend that should be included in the implementation planning?

Respondent:
To ensure the transition is as smooth as possible and does not cause downtime or disruption to normal operations, I believe it is crucial that the implementation takes place in clearly defined phases, with an emphasis on preparation, testing and a controlled ramp-up. The first phase should be a thorough analysis and validation of the requirements - i.e. confirmation that the system really corresponds to what individual departments need. This should be immediately followed by configuring the basic system settings in a test environment, where we verify that the designed workflow, permissions and interfaces correspond to practice.
The second key phase is pilot operation – i.e. deployment of the system to a selected department or type of requirement, where harsh testing will take place in real operation, but with increased supervision and the ability to respond quickly to comments. This phase will allow to reveal practical shortcomings without the risk of widespread failure. After the pilot, the comments should be evaluated and incorporated, and only then the system should be fully deployed across the company.
Planning should also include a plan for migrating data from the old system – including deciding what to transfer and what to archive. And of course, thorough user training before the live launch, setting up support during the first weeks, and a communication campaign plan for users to know what is changing, when, and why. The whole process should be accompanied by a clear schedule, delineation of responsibilities, and a crisis scenario in case major problems arise at some stage.

Interviewer (Participant):
Thank you for the detailed description. Can we also focus on how the new system should communicate with other systems in the company? What integrations do you consider necessary for the new approval system to fit effectively into your IT ecosystem?

Respondent:
Certainly, this is a very important topic. It is absolutely essential for us that the new approval system is fully integrated with our ERP system - especially due to the connection to budget data, accounting codes, centers and cost item codes. Ideally, the system should automatically verify whether the request corresponds to the budget plan and, after approval, should be able to pass the data directly to the ERP for further processing (e.g. generating an order or reserving funds).
Another important integration is with a document management system (DMS), because most requests are accompanied by contracts, offers or other attachments that we want to have stored centrally and traceably - not just inside the approval system. Two-way synchronization is important here, so that documents have uniform versions and can be accessed from both the DMS and the request perspective.
Furthermore, the system should be connected to our identity management – for SSO and user management. If a user leaves the company or changes position, this should be automatically reflected in the approval workflow. And no less important is the connection to the email system and calendar – both for notifications, but also for automatic holiday recognition and the ability to delegate approval.
It would be a great advantage if the system could also communicate with BI tools like Power BI or Qlik to create custom insights and reports. And in the future, I would appreciate an

open API so that we could eventually program our own integrations or functionalities as our needs evolve.

Interviewer (Participant):
Thanks for explaining the integrations. If we were to look at the long-term maintenance and development of the system, what would you consider to be the key factors that should be taken into account to keep the system up to date and effective over time?

Respondent:
In the long term, it is crucial for me that the system is not static, but allows for continuous development without the need for extensive intervention from developers. This means that it should have adjustable parameters – such as approval workflow, permissions, code lists, notifications or forms – that can be adjusted by a trained administrator from the company. If we need development for every little thing, the system will quickly become outdated or no longer meet current needs.
A transparent roadmap from the vendor is also important – we want to know how the system will evolve, what features are planned, and ideally have the opportunity to participate in this, for example through feedback or community voting. Support should include regular updates that do not require long downtimes, and the system should be modular enough to easily add new features or connections.
From a company perspective, I would like to see a clear plan for system maintenance – who manages it, who is the internal guarantor, how change requests are recorded, how their impact is evaluated and how new functions are tested. And of course, regular security reviews, access rights checks and capacity planning must be ensured – so that the system can handle potential company growth, new departments or process changes without unnecessarily redesigning the entire solution.

Interviewer (Participant):
Thank you. What factors would you consider to be most important in ensuring user satisfaction and acceptance of the new system? How can we ensure that the system will be effective and user-friendly in the long term?

Respondent:
I consider it absolutely essential for user satisfaction that users see the benefit in the system – that is, that it actually makes their work easier, not more complicated. This means that the system must be easy to use, logical, visually clear and responsive to common needs – no unnecessary extra steps, no technical obstacles that will discourage users. It is important that even a regular user without much IT skills can create a request, monitor its status and respond to comments or changes – without half a day of training.
In addition to the design itself, the availability of support is also key – that is, knowing who to contact if something goes wrong and being sure that the problem will be resolved quickly. This significantly affects the first impression and willingness to use the system. However, user acceptance is also a question of communication – it is necessary to explain in a timely and understandable manner what is changing, why, and what they will get out of it. Involving users in testing and collecting feedback before the live launch gives people the feeling that they are being thought of, and at the same time helps to fine-tune practical shortcomings.
And if we want the system to remain user-friendly in the long term, it is necessary to evaluate it regularly – ask what works and what doesn’t, monitor the use of features, collect feedback from real users, and continuously adjust the system as the company’s needs change. User-friendliness is not a one-time feature – it is something that must be actively maintained.

Interviewer (Participant):
Thanks for the great overview. What metrics would you consider key to measuring the success of the system implementation? How would you like to monitor whether the system is actually delivering the expected improvements in processes and user satisfaction?

Respondent:
Thank you, that is a very important question. I would measure the success of the implementation using both quantitative and qualitative metrics. Key quantitative metrics include, for example, the average time it takes to approve requests before and after the new system is deployed – if this time is significantly reduced, it is a clear signal of improved efficiency. I would also monitor the number of requests processed without errors or the need for intervention outside the system, or the number of internal escalations and support interventions – their reduction would indicate better usability and stability.
I would also like to track the adoption rate of the system – i.e. how many users are actually using the system, how often, and whether they are using all key features (e.g. comments, delegation, dashboards). Another metric could be the number and type of user queries or incidents during the first months – if they are repeated, it is a signal to improve the user interface or training.
At a qualitative level, I would recommend conducting a user satisfaction survey – a short questionnaire focused on ease of use, benefits to their work and overall trust in the system. It should be evaluated not only after launch, but also after a few months to see the development over time. And it is important to also monitor feedback from key users – i.e. managers who use the system to make decisions. If they see the benefit in the clarity and availability of data, this is a good indicator that the system is fulfilling its role strategically, not just operationally.

Interviewer (Participant):
Thank you for the detailed answer, it gives a clear idea of how to track the success of the implementation. Can we further focus on how you want the system to handle changes or updates over time? What change management method would you consider the best?

Respondent:
Absolutely. From my perspective, it is crucial that change management in the system is controlled, transparent and predictable – not ad hoc interventions according to the current need, but a structured process with clear rules. Ideally, I would like to have a central backlog of change requests where individual departments or users can submit their suggestions, and this list should be regularly evaluated in terms of priority, impact and complexity.
Every change should undergo an impact analysis – both technical and process – and should be communicated to the affected users in advance. It is also essential to be able to test changes in a test environment before deploying them to production. Ideally, changes would be deployed in the form of releases or packages in pre-planned waves, not unexpectedly during operation.
It is also important that after each change, revalidation takes place - i.e. verification that everything works correctly and that the change has brought the desired effect. In the case of major modifications, additional training should be provided or at least an update of the user documentation. The entire process should be led either by the system product owner or the IT coordinator in cooperation with key representatives of individual departments, including us from finance. Change management set up in this way ensures that the system remains alive, but at the same time stable and trustworthy for users.

Interviewer (Participant):
Thank you. What specific steps or tools would you consider key to ensuring a smooth transition between the old and new system? How should the new system support users during this transition period?

Respondent:
Thank you, this is the phase that will decide whether users will accept the new system or not. I consider a well-prepared migration plan to be absolutely crucial – not only technical, but also organizational. The first step must be thorough data preparation – i.e. cleaning and structured transfer of relevant requirements, budget codes, user accounts and history to the new system, if it makes sense. It must be clearly defined what will be migrated and what will remain in the archive of the old system.
It is also essential to have a period of concurrent operation or at least a sufficiently long pilot where users can safely experience the new system before the old one is completely shut down. During this period, direct user support should be available – i.e. hotline, chat or the presence of "local champions" who will help directly on site. The new system should actively guide users – in the form of an interactive guide, in-system help, tooltips, and ideally also by automatically notifying them of new features.
It's also important for a smooth transition for users to have access to comparisons – for example, an overview of where they previously found a certain type of request and where they will find it now. It also helps to create a set of short videos with typical scenarios, such as "How to submit a new request", "How do I find the approval status", "What to do if I need to delegate".
And last but not least – communication. Users need to know in advance when the transition will take place, what will change, what we expect from them and where to turn with questions. If we manage this well, people will not be afraid of the new system – on the contrary, they will see it as an opportunity to simplify their work.

Interviewer (Participant):
Thank you for the detailed overview of the transition process. What specific technical requirements should the new system meet from the developers' perspective to be stable and scalable? What should the development team take into account when developing the system to be ready for company growth and process changes?

Respondent:
I'm happy to answer. From a technical point of view, the new system should be built on a modern, modular and highly scalable architecture - ideally as a cloud or hybrid application that allows for growth without major performance limits. It is important that the system is built on a microservices or at least modular architecture, where individual parts (e.g. workflow engine, notifications, approval interface, reporting) can function independently and be easily expandable or replaceable.
The development team should anticipate that processes in the company will change from the beginning – the system should not be rigidly tied to one type of workflow. It should support configurable approval routes, custom fields, code lists, rules and roles – preferably via the administration interface. The data model should be flexible and extensible, with support for versioning of approval processes and their traceability.
Another key thing is a well-documented and secure REST or GraphQL API for easy integration with other systems. Developers should address robust logging, monitoring and alerting from the start – so that problems can be easily detected and responded to before they affect users. From a maintenance perspective, it is important to have a system ready for

CI/CD processes and testability – i.e. coverage with automated tests, the ability to run it in a test environment and easily move between versions without compromising data or availability.
And last but not least – the system should be optimized for performance to handle dozens to hundreds of simultaneous users, and ready for multilingualism, different time zones, and possibly localization for other countries if the company grows outside the Czech Republic. This will ensure long-term sustainability without the need for major technical redesigns.

Interviewer (Participant):
Thank you, and what specific security requirements should developers keep in mind when developing a new system to ensure it is sufficiently secure while also allowing for easy access management and protection of sensitive data?

Respondent:
Thank you, this is a very important area that cannot be taken as something "after the fact" - security must be part of the design from the very beginning. Developers should first and foremost implement robust authentication and authorization mechanisms. This means full support for SSO (e.g. via Azure AD, LDAP or another identity provider), with the option of two-factor authentication, especially for administrative and sensitive roles. The system must allow for role-level access management, but also finely define permissions according to document types, organizational structure or approval steps.
When working with sensitive data, encryption is essential – both in transit (TLS/HTTPS) and at rest (e.g. encrypting database fields, sensitive documents, access tokens). Furthermore, developers should implement protection against common attack types – such as CSRF, XSS, SQL injection, session hijacking – and perform regular security testing, ideally including penetration testing before going live.
Great emphasis should be placed on the audit trail – the system must record who created, approved, rejected, modified or deleted what, including time and IP address. Logs must be inviolable, available for audit and with the option of export. It is also advisable to implement detection of suspicious activities (e.g. multiple rejections within a short period of time, repeated access attempts) and have a mechanism for response – for example, automatic notification of the system administrator.
For access management, it must be possible to quickly and easily deactivate a user account (e.g. when an employee leaves) and ideally also connect to the company's HR or identity management system so that a change in position or termination of employment automatically switches permissions. Last but not least, the system must be designed to be GDPR compliant - i.e. allow for the management of personal data, control of access to it, and, if necessary, anonymization or deletion of data. Developers should follow the principles of "privacy by design" and "security by design".

Interviewer (Participant):
Thank you, and one last question. What features should the new system have to easily monitor its performance, availability, and overall health, and how should the development team approach these aspects during development?

Respondent:
It's a good thing. When it comes to performance and availability monitoring, the new system should have integrated tools or be ready to connect to common monitoring platforms (e.g. Grafana, Prometheus, ELK stack, Azure Monitor, etc.). The development team should

implement a set of metrics that will allow you to monitor both the technical status of the system and its operational health - for example:
Service availability (uptime) – monitor in real time whether key system components are available (e.g. API, database, frontend).
System response time – how quickly the system responds to user requests, especially for operations such as loading a dashboard, submitting a request, or approving.
Load (load, CPU, memory) – to detect system overload in a timely manner and adjust the infrastructure or optimize the code.
Number of active users and requests – monitor system utilization over time and prepare for growth or seasonal fluctuations.
Error rate – how many requests end in an error, broken down by type (e.g. 500, 403, 404) so that we can detect technical or process problems in a timely manner.
The development team should also implement central logging with full-text search capabilities – detailed logs of transactions, logins, system changes, and exceptions, ideally divided by level (INFO, WARNING, ERROR, CRITICAL). These logs should be readable, well-structured, and accessible to system administrators without the need for direct access to the servers.
From a process perspective, it is important to set up alerting – the system should automatically notify the system administrator or development team when a certain threshold is exceeded (e.g. API outage, high error rate or long responses). There should also be a status dashboard that shows in real time whether everything is working and where a problem may arise.
Developers should approach these aspects with a "build for observability" philosophy - not just writing functional code, but from the beginning, taking into account that the system must be continuously measurable, traceable and easily diagnosable. This ensures stable operation and quick response to potential problems, which is key to user and management trust.

Interviewer (Participant):
Thank you for the very detailed answer. With this complete overview, we have a clear idea of how the new system should work. If you have anything else you would like to add or clarify, I would be happy to hear it.

Respondent:
Thank you, I appreciate the space to add something else. Maybe I could just summarize one key thing from the finance department's perspective – the new approval system is not just a tool for "clicking off" requests for us, but a key element of company management. If it is well designed, it will help us better control budgets, manage costs, evaluate efficiency and catch potential risks in time. At the same time, if it is friendly and flexible, it will increase people's willingness to work with it correctly, which is absolutely essential for the quality of data and processes.
I would also like to emphasize the importance of a long-term partnership with the system supplier – not just a one-off implementation, but a truly living relationship where there is room for development, support and joint advancement according to the changing needs of the company. If this is achieved, the new system can bring not only efficiency, but also a culture of accountability and transparency throughout the organization.
Otherwise, you covered everything very thoroughly – thank you for your systematic approach. I would be happy to be a part of the next steps in the project.

Conclusion of the interview with participant 7

Interviewer: Thank you for showing us the interview sample. It was really interesting to see it in action. I would like to ask you a few questions now about your methodology and approach to requirements gathering. Can I?

Respondent: Sure, ask, I'll be happy to answer.

Interviewer: What development methodology do you use most often? Do you work more agile, traditional, or in some hybrid mode?

Respondent: I personally work exclusively within the traditional methodology – mostly classic waterfall. The projects I participate in have firmly defined phases: analysis, design, development, testing, deployment. These are mostly projects in larger companies, where there is pressure on predictability, documentation and approval processes.

Interviewer: And how do you proceed with collecting requirements within this methodology? Do you have a strategy for obtaining the most accurate information during the interview?

Respondent: Definitely. First, I prepare a basic framework of topics I want to cover – usually based on previous documentation, terms of reference or process model. In the interview, I first try to understand the context and role of the person – I ask about their daily work, where they spend most of their time, what problems they have. Then I go more in-depth – I use questions like "what happens when…", "how will you know it's done", or "what would happen if…".

Interviewer: So you combine open-ended questions with hypothetical ones?

Respondent: Yes, exactly. I try to avoid questions like "what would you like" because that often leads to a specific solution without understanding the real problem. I prefer to ask about current procedures, what doesn't work, what is time-consuming, what is often repeated. I also ask about exceptions – that helps a lot to reveal what is missing in the process.

Interviewer: And what types of requests do you try to get during the interviews?

Respondent: Most often functional requirements – i.e. what the system should be able to do, what steps the user should take, what outputs should be created. But within the traditional methodology, we often also cover non-functional requirements – for example, requirements for system availability, responsiveness, access rights, audit trails, etc. I usually solve technical requirements only in cooperation with the architect or developers.

Interviewer: Do you write the requirements yourself, or do you collaborate with someone else?

Respondent: I develop the initial draft of requirements myself – based on interviews, process documentation and possibly observations. Then I validate them with the respondents – either in the form of a meeting or a document review. And before the requirements are forwarded to the developers, they are approved by the business owner and sometimes the architect.

Interviewer: What format do you use for requests? Do you have a standard template?

Respondent: Yes, we have an internal template that contains an identifier, description, input, output, rules, exceptions, and related processes. For more complex requirements, I also prepare scenarios and use cases. In some companies, I also added wireframes or screen designs - especially when it was a new system.

Interviewer: How do you know when your requirements are sufficiently complete?

Respondent: If I have all the main scenarios covered, including exceptions, and if users don't introduce new additions or questions during validation, I take that as a sign that we are close to the goal. In traditional environments, we also often hold that it must be possible to prepare test scenarios based on the requirements - if not, the requirements are not clear enough.

Interviewer: Does it happen that some requirements change during the course of the project? How do you deal with that?

Respondent: Yes, it happens, although in a traditional environment it is not entirely welcome. We have to officially record changes – usually with a change request. This is approved and only then are the specifications updated. This often causes delays, so I try to prevent this by asking a lot about exceptions and scenarios right from the start.

Interviewer: Thank you, that's very detailed. How often do you use observation or shadowing instead of interviewing?

Respondent: Quite often, if the project allows it. Sometimes people don't even realize what they're doing exactly – or they can't describe it accurately. When I watch them at work, I often notice unnecessary steps or informal detours through the system that wouldn't come up in an interview.

Interviewer: And when you're collecting requirements, how do you work with technical roles – like IT or architects?

Respondent: We usually organize separate meetings there, where I present them with proposals from a business perspective. They then give feedback on what is technically feasible and what is not. Technical requirements such as integration, security, performance – I leave that to them to elaborate in detail, but I try to at least understand it in general terms so that I can reconcile both sides.

Interviewer: Thank you very much, that was a very useful part of the interview.

Respondent: I'm glad it happened.

Participant 8

Beginning of the interview with participant 8

Interviewer: Hello, thank you for taking the time to talk to me. First of all, how are you today?

Respondent: Hello, thank you for asking, I'm fine. It was a bit hectic this morning, but things are calm now. How about you?

Interviewer: Okay, thank you. So we can start with a few questions. How long have you been working as a business analyst?

Respondent: I've been doing this for over twenty-five years. To be exact, since 1998, so it's really been quite a long time.

Interviewer: That's really impressive. How did you actually get this role?

Respondent: Well, I originally started as a project coordinator. I gradually got more involved in proposing solutions and communicating with clients, until I naturally started to profile myself more as an analyst. Back then, roles weren't as distinct as they are today, so it was kind of fluid.

Interviewer: So you had a different position before?

Respondent: Yes, the project coordinator. And at the very beginning I was doing administrative support for a small IT team, so I really started from scratch.

Interviewer: What types of projects do you have the most experience in? I'm thinking IT, process analysis, strategy...

Respondent: I worked mostly on IT projects, but always with an overlap into process analysis. It was often about making the company's operations more efficient, so in addition to system design, also understanding internal processes. Strategy as such was more marginal.

Interviewer: And have you worked in more than one company, or do you have long-term experience in one organization?

Respondent: It was four different companies, but I always spent a longer time there, on average around six to eight years. For me it was always a question of stability and good relationships in the team.

Interviewer: And were they more likely larger companies or smaller ones?

Respondent: Rather smaller. They all had up to twenty-five employees. I enjoy an environment where you know all your colleagues and have the opportunity to really influence things.

Interviewer: I understand, thank you. That environment certainly brings a lot of flexibility. Thanks a lot for sharing, so we can start the experiment that we have prepared. Here is your assignment and once you are ready, we can start.

Participant-Respondent Interview (ChatGPT)

(https://chatgpt.com/share/67ec3d72-1ac4-8008-a0f8-f2c18551142b)
Interviewer (Participant):
Hello, today I would like to discuss your view of the current approval system with you, what you dislike about it, what you would like to keep, and what the new system should definitely be able to do. Let's start lightly. Could you tell me how you actually use the approval system in your department in practice now?

Respondent:
Hello, of course. In our department, we use the current approval system primarily to approve purchase requests, budget changes, and internal financial documents, such as invoices or internal cost transfers between centers. In practice, the request is created by the relevant employee or center manager, entered into the system, and then sent to the individual responsible persons for approval according to the preset workflow - most often at several levels, e.g. the center approver, then the finance department, and sometimes management. We in finance then check the correctness of the amount, budget coverage, and compliance with internal guidelines, if applicable. We then send the approved requests to accounting or purchasing. It works, but we often encounter limitations, which I will get to in a moment.

Interviewer (Participant):
Thanks, that sounds like a pretty robust process. Let me break it down a bit. When you say you're encountering limitations, what do you think is the biggest problem with the current system, what hinders or frustrates you the most in your daily work?

Respondent:
Perhaps the biggest problem is the inflexibility and complexity of modifying approval workflows. When the organizational structure, responsibilities, or just need to add a new approver changes, we have to deal with it through the IT department and wait several days for the change to be reflected in the system. In practice, this means that approvals are stalled, requests hang without movement, and people contact us asking why it is taking so long. In addition, the system cannot respond to specific situations - for example, when someone is unavailable (vacation, illness), it cannot automatically redirect approval to a representative. We have to deal with it manually, which leads to errors and delays.

Interviewer (Participant):
I understand, it must be quite frustrating. And when you mentioned approval during absence. Do you have at least some mechanism for delegation or substitution in the system now, or is it purely a matter of someone having to manually reset or urge it?

Respondent:
Unfortunately, there is no automatic or user-friendly mechanism for delegation in the current system. When someone is absent, they either have to arrange it in advance through IT – which most people don’t do, or we in finance find out who is the substitute and have to manually bypass the system, send emails or approve “outside the system”, which is of course risky and not entirely auditable. And when the absence is unexpected, such as illness, the process gets completely bogged down. So yes, this is one area where the new system should clearly be able to flexibly and automatically handle substitution, ideally based on pre-set rules or calendar availability.

Interviewer (Participant):
I understand, that sounds like something that really complicates normal operations. Let's look at it from another angle. How do you actually know in the current system that a request is stuck? Do you have an overview of where it got stuck, or do you only find out when someone complains that it's taking too long?

Respondent:

Unfortunately, in most cases, we only find out when someone complains that it's taking a long time - for example, the head of the center calls us to say that his request has been "hanging around for a week" and wants to know why. The system has basic overviews, but they are not clear or intuitive enough for the finance department to have a quick overview of all requests in the approval process, let alone where they are. There is no central display like "approval queue by status and person" that would show us how many requests are waiting for whom. So we often have to search for each request manually, which costs us a lot of time. The new system should definitely have a quality dashboard with the ability to filter requests by status, responsible person, department, etc., so that we can quickly identify bottlenecks.

Interviewer (Participant):
That sounds like a lot of manual work. And speaking of reports. Besides tracking held requests, what else would you appreciate in dashboards or reports? Are there any specific metrics or insights that you are currently missing and have to search for or calculate on the side?

Respondent:
Certainly yes. Now we have to do most of these reports manually in Excel – for example, when we want to know how many requests were approved in a certain period, how many were rejected, the average approval time, or how long it takes individual approvers to express their opinion. These are important metrics for us, because they are the basis for evaluating the efficiency of processes and looking for areas where things are going wrong. Ideally, we would like to be able to generate these reports automatically in the new system, ideally with the option of exporting to Excel or Power BI. I would especially appreciate reports by type of request (purchase, budget, invoice), by department and even by approvers – who is delaying the most, where the bottlenecks are. We are missing this now and without it, it is difficult for management to argue when we need to change something in the processes.

Interviewer (Participant):
That makes sense, such metrics are key when you want to improve processes. Let's go back to the system users for a moment. How would you describe the level of regular users when it comes to working with the system? Do you feel that they are familiar with it, or do they often fumble and turn to you or IT with questions?

Respondent:
Most regular users – by that I mean, for example, center managers or administrative staff – are only partially familiar with the system. They understand the basic procedure for submitting a request, but as soon as they need to edit something, look up the approval history or find out why something costs, they start to fumble. And because the system is not very intuitive, they often call our finance or IT department directly. In addition, there is no proper help or guide, so we have to train new colleagues ourselves, which slows us down. If the new system had a clear and simple user interface, ideally with context-sensitive help or a guide for individual steps, it would help significantly. At the same time, it would reduce the number of routine questions that we now have to deal with.

Interviewer (Participant):
Thanks, that's a very valuable insight. When you talk about training new people. Do you have any formal training or documentation that you give them now, or is it more of an informal process of sit down with me and I'll show you?

Respondent:
It's more the latter case – an informal process. When someone new joins us, a more experienced colleague usually takes them and shows them how to submit applications and what to click on. We don't have any official documentation or training materials, because the current system has been patched up over the years and the documentation either hasn't been created at all or no longer reflects reality. So everything works based on oral transmission of know-how. It's obviously inefficient and depends on the specific person. I would definitely like the new system to include simple user documentation, ideally video tutorials or interactive guides, and the ability to easily train new people without the need for personal transmission of experience.

Interviewer (Participant):
I understand that when it's up to individuals, it can easily become a bottleneck. And now let's look at the communication regarding approval. How do users currently receive information about what's happening with their request? Are notifications working well, or do you feel there's room for improvement there too?

Respondent:
Notifications are another weak point of the current system. They only work partially – an email is sent when a user has to approve something, but if someone else approves or rejects the request, the original requester no longer receives any information. This leads to people having no idea what stage their request is at, and they often call us saying "they submitted something, but nothing has happened since then". Moreover, emails are not centralized in any way – if a user overlooks or deletes them, they have no overview in the system of "what is waiting for my approval" or "the status of my requests". The new system should clearly have a sophisticated notification mechanism – both by email and within the system interface itself – with the possibility of user settings, e.g. daily summary, immediate notification, reminder after X days, etc. And of course, it should also have a personal dashboard where everyone can clearly see the status of their requests.

Interviewer (Participant):
That makes sense. And as for comments or notes on requests, do you now have the option to comment on something in the system, to respond to other people's comments, or is it handled outside the system, for example by email?

Respondent:
Unfortunately, it is mostly outside the system now. Although there is a basic field for a note when you create or approve a request, it is very limited – it is not clear, you cannot respond to it, there is no notification that someone has added something. So when we need to clarify something or ask a question, we still write emails or call. However, this means that the information is not stored together in the system and when we return to the request after some time, we do not have the complete context. In the new system, I would very much like the ability to add comments directly to the request – ideally in the form of a thread, so that it is clear who is responding to what – and for the system to send a notification about it. This would significantly improve the transparency and traceability of communication.

Interviewer (Participant):
This is a great point, the context of requests is often missing when everything is handled sideways. And if we delve a little more into security. How important is the audit trail for each

request to you? Do you now track who changed what in the system and when? Or is it another thing that you are missing there?

Respondent:
Audit logs are crucial for us, mainly for control and compliance with internal guidelines and also for auditing – both internal and external. The current system does keep some logs, but they are very basic and difficult to access – we often have to obtain them through IT and they are not presented in a clear way. For example, we cannot see when someone performed a specific step, whether they just opened something or actually approved it, and who made any changes to the request after it was submitted. Ideally, the new system should keep a detailed audit trail – including time, user name and description of the action – and it should be a standard view available to us as administrators or the control department. This is also important so that any disputes or ambiguities in the process can be resolved objectively.

Interviewer (Participant):
Of course, without a quality audit trail, it is difficult to trace what actually happened. How important is it for you that the system is connected to corporate login, for example via SSO? Do you have any specific security requirements that the new system should definitely meet?

Respondent:
We consider the connection to corporate login via SSO to be essential. First, for security reasons – so that we don't have to deal with additional passwords and access data, and secondly for authorization management, which is already centralized via our identities in Active Directory. We want access to the system to be governed by user roles and the organizational structure that we have defined in our IAM system. At the same time, it is important for us to be able to audit accesses at the SSO level. In terms of security, we expect support for two-factor authentication, logging of all accesses, a regular backup mechanism and, of course, data encryption – both during transmission and at rest. Protection against unauthorized access to sensitive documents must also be ensured, especially for contracts or financial budgets.

Interviewer (Participant):
I understand, so your security and access control is really at a high level. And as for the access rights themselves. Do you need them to be configurable in a very detailed way? For example, different departments will only see their own requests, different levels of rights to read, edit, comment, etc.?

Respondent:
That's right, detailed access rights settings are absolutely key for us. We need to be able to manage rights not only by department, but also by specific roles – for example, the applicant has different rights, the approver has different rights, and the system administrator or controller from the finance department has completely different rights. We want, for example, the head of the center to see only the requests of his team, but not to see documents from other departments. At the same time, we also need adjustability at the level of individual steps in the process – someone can only read, someone can approve, someone can edit attachments, and someone can only add comments. It should also be possible to determine who can apply for which types of items – for example, investments or services above a certain amount. And what is important – all this should be easy to manage without the need for IT intervention.

Interviewer (Participant):
That makes perfect sense. And when we look at the process side, do your approval scenarios change often? For example, do some requests have a different approval path depending on the type, amount, or perhaps the center? How complex are your workflows?

Respondent:
Yes, our approval scenarios change quite often and are quite complex. For example, we have different workflows depending on the type of request – a different process applies to a purchase request up to 50 thousand and another over 500 thousand, or special approval for investments, services, or expenses from project funds. The organizational structure also comes into play – some departments have specific rules, for example, everything above a certain amount must be seen by the operations director. We also have specifics for internal transfers or estimated items. And as I mentioned, when the organizational structure or responsibilities change, we have to adjust the workflow – which is painful in the current system. In the new solution, we would definitely like to be able to easily set up and manage different scenarios – ideally using some visual tool where we can define branching by amount, type, center, or even project number. Flexibility in this direction is essential for us.

Interviewer (Participant):
That sounds like a very dynamic environment. And I'm wondering, do you need the system to handle any exceptions or escalations? For example, when something is not approved on time, or when a step cannot be completed for some reason?

Respondent:
Yes, this is exactly what we are missing in the system today and it often delays us. We need the new system to be able to work with exceptions and escalations. For example, if someone does not approve a request within a certain period of time – say 3 days – the system should automatically notify the person concerned, or escalate the request to a superior. We would also like it if it were possible to skip or delegate the approval step when the approver is unavailable for a long time, of course with an appropriate justification and audit trail. And in some cases we also need the ability to manually trigger an exception – for example, when an expense outside the regular budget needs to be approved quickly, but with the knowledge of management. Ideally, this should be part of the process definition – i.e. we can set rules for when and how exceptions or escalations work.

Interviewer (Participant):
Great. Now a little differently. Do you now have any connection of the approval system to other company tools? For example, to ERP, accounting, or a document system? And if so, how does it work and what would you like to keep or improve in this regard in the new solution?

Respondent:
Yes, we have a partial connection to the ERP system – specifically our accounting and budget system. In practice, this is how it works: after the request is approved, the selected data (e.g. center number, cost type, amount, note) is transferred to the ERP, where the corresponding record is created – for example, an internal order or budget reservation. But it is quite cumbersome – the integration is not two-way and it is technically quite fragile. If something changes or breaks somewhere, often no one will recognize it right away. Moreover, in the current system, we cannot see retrospectively whether and when the request was transferred to the ERP, which complicates the search. For the new system, I would definitely like to have

a robust and two-way connection to the ERP – ideally via API – and a clear control of data flows. And if the system could also handle integration with a document repository or DMS, where, for example, attachments to requests, contracts or invoices would be stored, that would be a huge plus. Now we deal with it outside the system and the continuity is lost again.

Interviewer (Participant):
Thanks, this is very specific. When you mentioned attachments. How do you actually work with documents attached to applications now? Are they stored directly in the system, or are they sent outside? And how do you think attachment management should ideally work in the new system?

Respondent:
In the current system, attachments can be attached directly to the application, but it is very limited – both in size and format. Moreover, they are not organized in any way – you simply attach a file and that's it, without metadata, without versioning. When someone uploads a new version of a document, the old one is overwritten and often the overview of what was originally there is lost. And there is a complete lack of control over who has access to which attachment – if someone has access to the application, they automatically see all the documents, which is not always desirable, especially for sensitive attachments such as offers, contracts or budget documents.
Ideally, the new system should allow for clear management of attachments - with the option of versioning, marking the document type (e.g. offer, budget, contract), the ability to comment on a specific document, and most importantly, the ability to manage access rights even at the level of individual files. A big plus would be a connection to the DMS or the company's central repository, so that documents are not duplicated and are searchable even outside the approval system, but still linked to a specific request.

Interviewer (Participant):
I understand, so it would require more than just an attachment basket. More like living documentation for the application. Now I'll ask from a slightly different angle. What do you think the form itself for submitting a new application should ideally look like? Do you have any ideas or requirements for fields, structure, validations, or perhaps help?

Respondent:
Exactly – live documentation, not just "I'll paste a PDF and that's it". As for the application form itself, I would have a few clear requirements. First of all, it should be clear, dynamic and adapt to the context – i.e. only relevant fields would be displayed depending on the type of application (e.g. purchase, budget change, internal transfer). It makes no sense for the user to fill in ten items when three is enough for a given type. Furthermore, the form should contain pre-filled values where it makes sense – for example, the organizational unit according to the logged-in user, or automatically offer the budget center according to their classification.
Validation is also important - for example, checking the amount format, filling in mandatory fields, warning about exceeding the budget before sending, etc. And there must be help directly next to the fields - for example, in the form of a tooltip or a small note - so that the user knows what is expected of him. Last but not least, I would appreciate the option to interrupt the entry and save the proposal as a draft - this is not possible today and if you "get lost" halfway, you have to start over. Simply an intuitive, smart and contextual form, not just a digital copy of a paper form.

Interviewer (Participant):
Great, this is a great idea. And if we go back to budgets for a moment. You mentioned that the system should warn you if you exceed the budget. Do you have any rules that the new system should automatically check? For example, spending against the plan, limits by type of expense, or something similar?

Respondent:
Yes, we have several rules that we would like to check automatically in the new system, because today we do it manually, or not at all. The basis is drawing against the approved plan - each request should be checked against the current budget status of a specific center or project. If the request exceeds the planned amount, the system should either warn or automatically move the request to a higher level for approval. We also have limits according to the type of expense - for example, the purchase of small materials has a monthly limit per center, while services have a limit per individual supplier. The system should be able to evaluate these limits in real time.
Another rule is, for example, checking for duplicates – typically when something is requested that has already been approved by another department, or when one request is broken down into several smaller ones to "pass under the limit". Even there, the system should be able to warn or require additional approval. And the last thing – checking compliance with internal guidelines, e.g. that a certain type of expense must have an appendix with three offers or that a contract over a certain amount must be attached. Ideally, the system would take care of this automatically when entering.

Interviewer (Participant):
That's great. And if we think about how the entire life cycle of a request works. What would you like to have available when you revisit a request? What information should be there so that you have a good overview of what happened and why?

Respondent:
When I go back to an old request, I need to see its entire "story" - from submission to final output. This means a clear and chronological overview of all steps: who submitted the request, when, what was written in it, who approved or rejected it, with what comment, and if there were any changes to the content (e.g. amount, budget item). Ideally with time stamps and the exact name of the user. Furthermore, there should be all attachments, including version history, and of course an audit trail of all changes - who modified what when. I also want to see what the result was - e.g. that the request was transferred to the ERP as internal order No. XYZ and when this happened.
And for me, visibility over the context is absolutely essential – that is, why something was approved or not approved, what led to it, what the communication was like. If there is a well-conducted discussion in the form of comments or notes, it will help significantly. Today, we often look it up from emails, which is time-consuming and inaccurate. Ideally, the entire request would function as a "journal" of the given request – clear, complete, and easy to find.

Interviewer (Participant):
I understand. Now a little about the pilot deployment. When the new system is ready, do you imagine that it should be deployed to just one department or type of request at first? Or do you want a hard start for everyone?

Respondent:

I would definitely prefer a gradual deployment – ideally a pilot in one department and on a specific type of request, for example purchase requests. This will allow us to verify functionality, debug processes, collect feedback from users and, most importantly, catch any errors or shortcomings in time before the system is expanded to the entire company. The finance department should definitely be actively involved in the pilot, because we have an overview of processes across the company and can catch discrepancies in time. A sharp start for everyone would be risky in my opinion – we would overload support, there would be confusion among users and it could unnecessarily damage the system right from the start. A gradual ramp-up with a well-managed pilot is a safer path.

Interviewer (Participant):
That's a perfectly reasonable approach. And what do you think user training should look like before a live launch? What worked for you, what would be ideal for different groups of people: regular users, approvers, admins...?

Respondent:
We know from experience that one-size-fits-all training doesn’t work for everyone – each group has different needs and levels of knowledge. It’s ideal to divide training by role: regular users should receive brief, practically focused training – ideally in the form of a video or an interactive guide directly in the system, where they will learn how to enter a request and track its status. Approvers also need to know how to work with comments, how to respond to exceptions or how to enter delegation. And administrators need to go through detailed training – not only on how to manage permissions, but also how to set up workflows, check integrations and handle exceptions.
It has proven to be effective to have short instructions or checklists for each group, as well as the possibility of contacting one of the internal “superusers” or ambassadors. A testing environment is also ideal, where users can try everything out without risk. And most importantly – training should not be a one-off, but should also be available retrospectively – in the form of an intranet signpost, video tutorials or contextual help in the system. This helps a lot when training new people and when solving common questions.

Interviewer (Participant):
That's really thoughtful. And speaking of users, how would you imagine collecting feedback after launch? Do you have any idea how that should work so that the insights can be quickly used to improve the system?

Respondent:
Ideally, feedback collection should be a combination of several channels. First, I would definitely like to have a quick evaluation option built directly into the system - perhaps in the form of a short questionnaire after completing an application or approval, where the user could check if the process was understandable, if something was missing or what delayed it. This should be simple and unburdening so that people actually use it.
I would also suggest setting aside the first month or two after launch as an official “feedback period” – with a dedicated channel for reporting issues and suggestions (e.g. a shared Teams group or email inbox) that someone responsibly manages and actually responds to suggestions. We have also found short face-to-face interviews or quick online questionnaires for select groups of users – especially center managers or approvers who have more comprehensive experience.
And what is essential – feedback must not end up in a drawer. I want to be able to discuss feedback with IT and the system supplier, prioritize it and have a clear plan for what will be

changed and when. Ideally, a visible changelog or overview of “what is coming” so that users can see that their comments are having an impact. This greatly helps with system adoption.

Interviewer (Participant):
This is a great approach. Now I'm curious, who do you think should be the key contact for users after the system is launched? Do you have someone specific in mind, or would you like it to be handled by a helpdesk or a superuser in each department?

Respondent:
The ideal model is a combined one – a central helpdesk for technical matters and a parallel network of trained superusers in individual departments. The helpdesk would solve general problems with the system, login, errors, etc., while the superusers would be the first point of contact for process questions – e.g. “what request should I use”, “what if I need an exception”, or “why is this person approving this in our company”. There should be someone in each department who knows the processes, is a little more technically savvy and can filter or resolve common queries without unnecessary escalation.
For the finance department, I would definitely appoint someone as a “process guarantor” – that is, someone who will ensure that the approval logic complies with the rules, collect feedback from the department and coordinate any changes. This person should also have a direct line to the system administrators and IT. It is important that users know who to contact with what, and are not frustrated by their query being bounced between three people. So, in my opinion, a clear division of roles and responsibilities from the start is key.

Interviewer (Participant):
That sounds good. I'm also curious about how you would imagine managing changes in the system after launch? When it turns out that something needs to be adjusted in the workflow, rule, field. What process do you think should be ideally set up to make it transparent and manageable?

Respondent:
Ideally, I would imagine that changes to the system after launch will be managed by a formal, yet flexible process. The key is that changes are not made ad hoc, but based on validated requirements. This means that each change request – whether from user feedback or operational need – goes through one place, for example a designated process manager or a smaller coordination group (ideally consisting of finance, IT and key departments).
This team would assess the impact of the change – whether it is just a technical adjustment or has a wider impact on processes, security or reporting. The change would then be included in a change plan – ideally with a basic description of why it is being done, who is responsible for it and when it will be deployed. Changes should be accumulated in regular waves if possible – for example once a month – so that the system does not become a constantly changing target. And I would also greatly appreciate it if each change was announced in a clear way to users – for example in the form of a brief newsletter or a “what’s new” notification. The key is that the right people know about the changes and that continuity is not lost. So, in my opinion, transparency, planning and communication are the three pillars on which this should be based.

Interviewer (Participant):
Perfect. And now one of the very last things. How will you personally evaluate whether the transition to the new system was successful? Do you have any specific criteria or indicators that will tell you yes, this was successful?

Respondent:
Certainly yes – I have a few clear criteria in mind by which I will know that it makes sense and the transition has been successful. The first is the speed and smoothness of the process – if I see that requests are processed faster, do not get stuck and users stop complaining that "it's hanging somewhere", this will be a major signal of improvement for me. The second criterion is user autonomy – if the number of routine queries to finance or IT is reduced, it means that they are familiar with the system and it is intuitive for them. The third indicator will be the quality and use of data – if we have meaningful reports, overviews and audit records that allow us to better manage budgets and processes, this is a great added value. And last but not least – feedback from the department. If the center managers or approvers say that they are working well with it and do not see it as a necessary evil, but as a tool that helps them, then I will know that we have set it up correctly. So success for me is not just technical commitment, but real improvement in everyday work.

Interviewer (Participant):
Thank you, this is a great conclusion. Thank you very much for your time and openness. Is there anything else you would like to add, or any specific request that we may not have mentioned that should definitely be included in the new system?

Respondent:
Thank you, I think we covered a lot and went into a lot of depth, which I appreciate. I might add one more thing that is often left out – and that is mobile access. Today, people, especially leaders and managers, often approve requests on the go or outside the office. Therefore, the new system should have a fully functional mobile version or, ideally, an application that allows for convenient approval, an overview of requests, and easy addition of comments from a mobile phone. There is no need for someone to create complex requests via mobile, but approval and communication definitely need to be there.
Otherwise, I would say that you have covered all the key areas. If the system can be designed according to these requirements, it will be a huge step forward. And thanks again for the opportunity to discuss these things openly - I believe that this will help to ensure that the new system really corresponds to what we need in practice.

Conclusion of the interview with participant 8

Interviewer: Thanks again for the sample interview, it was very inspiring. I would like to follow up a bit and ask you about your working approach. I could start by asking – what development methodology do you actually currently use in the company?

Respondent: We are currently working mostly in agile mode – specifically a combination of Scrum and Kanban depending on the type of project. In the past, I have also worked in purely traditional environments, mainly in projects that had a clearly defined scope, budget and deadline – typical waterfall. But for the last ten years, I have been working almost exclusively in an agile framework, because it brings much greater flexibility, more frequent feedback from users and better response to changing requirements.

Interviewer: And when you interview clients or users, do you have a strategy for how to get the most accurate and appropriate requirements from them?

Respondent: Definitely yes. My strategy is based on a combination of open questions, gradual immersion and modeling of situations. At the beginning, I try to establish trust – similar to what the two of us did. First, a general question: "How are you currently working with this process?", "What is most important to you in this area?" And only then do I move on to deeper questions like: "What happens if step X does not go well?" or "How will you know that it is working well?"
The order of the questions is also strategic – functional context first, then exceptions, then success metrics. And I use a lot of scenario questions like "Imagine that…" because people often don't know how to describe something theoretically, but they can do it very precisely in a story.

Interviewer: That sounds very thoughtful. And what types of requirements do you most often try to identify during these conversations?

Respondent: I always try to cover the whole set – functional, non-functional, technical and process. Functional requirements come up most often – what should the system do, how should the input look like, what actions should it enable. But non-functional requirements are also very important – they are often underestimated. I ask things like "How quickly do you expect the system to respond?", "How important is availability outside of working hours?" or "How important is it that the system can be used from a mobile phone?"
I then consult the technical requirements more with the IT team – typically around integrations, data formats, connections to identity management, etc. And then there are the so-called latent process requirements – you can only find out by observing or by asking about exceptions, workarounds, what users have to "somehow solve on the side". These are often key moments that decide whether the system will really work.

Interviewer: Do you have a system or framework for structuring your requirements during the interview?

Respondent: I use a simple table or canvas where I keep notes by category: "goals", "pains", "features", "exceptions", "bindings". Sometimes I even draw it – for example, a process flow or a screen as the user sees it. It helps not only me, but often the respondent himself, who becomes aware of connections that he did not notice before. And after the interview I do a quick sorting – what is a must-have, what is nice-to-have, what is unclear and will need to be supplemented.

Interviewer: That makes a lot of sense. Could I ask what specific questions have worked best for you when gathering requirements? Do you have any favorite wordings?

Respondent: Definitely. I use questions like this a lot:
"What happens if this step doesn't go right?" – this one will reveal exceptions. Then, for example: "How will you know you're doing this right?" – this will get me to metrics and the target state. Or: "How do you imagine the ideal state if nothing was limiting you?" – this one is great for revealing needs that they wouldn't otherwise say. And then I also like questions like: "What do you have to do manually or bypass today because the system can't do it?" – these often reveal the biggest pain points.

Interviewer: And when a request comes in, how do you know when it's specific enough? That you can take it and implement it?

Respondent: If I can imagine it in the system – ideally in the form of a user scenario. For example: “The user opens the form, enters the amount, the system shows them the budget balance and immediately checks the limits.” If it remains general like “it should be clear”, then I ask further: “What does clear mean to you?”, “What would you like to see there first?” or “What is holding you back the most right now that it is not clear?” – I try to get to the level of a specific interaction or expected reaction of the system.

Interviewer: And what if you encounter resistance? For example, if the respondent doesn't want to answer or just says "it doesn't matter"?

Respondent: This happens, especially when someone doesn't have time or is skeptical about the outcome of the project. Empathy helps there – just acknowledging: "I understand that this may be annoying for you, but your input will really affect how it will work." Or I change tactics – I move on to a lighter question, ask about their experience or a story from practice. When someone remembers a specific situation, for example, when the system failed or something upset them, they start talking naturally.

Interviewer: Do you have a way of checking that you have understood the requirements correctly? Do you do a recap, for example?

Respondent: Always. At the end of the interview, I summarize what I wrote down and say: “Can I check if I understood correctly?” Then I might reformulate the functional requirement and see if they respond with confirmation or correction. And sometimes I show them a diagram or a simple wireframe if we already have one – visualization helps a lot with accuracy.

Interviewer: Is there any advice you would give to junior analysts who are just starting out doing interviews?

Respondent: Definitely – for me the most important thing is not to be afraid of silence. When you are silent for a while after a question, people often think and say something valuable. Second thing – don’t jump into the solution right away. Leave space for the respondent to express what they want. And third – write down or record everything, because even an inconspicuous remark can later reveal a key requirement.

Interviewer: That's great. And the last question on this topic - how do you know when you have enough requirements and can move on to analysis?

Respondent: Mostly based on the fact that new conversations are repeating themselves – that I hear similar needs, similar problems. We call it saturation. When nothing new is added, but rather previous knowledge is confirmed, it is time to consolidate and start creating an analysis or proposal. But I never take it for granted – I leave room for additions if I encounter gaps during the proposal.