Software processes, quality, and standards (course and arrangement), 10.12.2022

TALLINN UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF SOFTWARE SCIENCE

Filmography website for the Ministry of Culture of Estonia

Maintenance activities and retrospective (excerpt)

Lab 4 in subject "Software Quality and standards" (IDY0204)

Authors, student group

Maksym Avramenko (223884IVSM) Developer

Viktoriia Abakumova (223890IVSM) Tester

Fillip Molodtsov (223891IVSM) Maintainer

Azhar Kazakbaeva (223909IVSM) Acquirer (contact person)

Team ID: 2+2

Presented: 10.12.2022

Supervisor: Jaak Tepandi

TALLINN 2022

A10. Proposing system related metrics and forecast. 3

10.1 Metrics. 3

10.2. Forecast. 5

A11. Implementing ISKE, ITIL, ISO 9001, or another framework (intro). 8

11.1. Framework selection. 8

11.2. Overview of implementation. 8

A12. Retrospective analysis, references, authors 10

12.1. Retrospective analysis. 10

12.2. Presenting the team 10

12.3. References 11

A10. Proposing system related metrics and forecast.

10.1 Metrics.

Production analytics are necessary measurements for this project. We chose two metrics: mean time between failures (MTBF) and mean time to respond (MTTR).

First of all, both metrics measure assess how well the software performs in a production context (environment). These metrics clearly show the current state of affairs and can be used for meetings with the client, as they are precisely expressed and clear for everyone to understand.

MTBF is known to be a crucial maintenance metric to measure performance, safety, and equipment design.

Knowing and understanding MTTR can help project team to react to measure a team's success in neutralizing system attacks (cybersecurity issues).

MTBF Calculation

The Mean Time Between Failures (MTBF) is the average time between repairable production failures. The metric is used to track both the availability and reliability of a product. The higher the time between failure, the more reliable the system. The goal is to keep MTBF value as high as possible. The following is the equation we are using to calculate MTBF:

$$\text{MTBF} = \text{Total Uptime} / \text{Number of Failures}$$

It means that MTBF is calculated using an arithmetic mean – taking the data from the period we want to calculate and dividing that period's total operational time by the number of failures.

As soon as the system we are working on is not in production, we will calculate the MTBF value for the QA environment. Our team receives all data while working on the previous laboratory work (testing).

The System is operational 24 hours a day, seven days a week, with 4 outages per month, the first one was lasting up to 25 minutes, the second up to an hour, and last two lasting up to 40 minutes.

Total Working Time: 720 hours which is 24 hours in 1 month

Number of Failures: 4

Outages Total Failure Time: 25 minutes + 60 minutes + 2*(40 minutes) =165 minutes = 2.75 Hours

(720 – 2.75) / 4 = 179.31

MTBF = 179

This means that when the system is in production, the average time between failures is rounding 179 hours which is a good value.

MTTR Calculation

MTTR (mean time to respond) is the average time it takes to recover from a system failure from the time when we are first alerted to that failure. This does not include any lag time in an alert system.

You can calculate MTTR by adding up the total time spent on repairs during any given period and then dividing that time by the number of repairs.

MTTR = Total time spent on repairs / Number of repairs

For this metric we used OWASP and revealed some potential security issues in the developed system.

Incidents: 6

Time spent on repairs: 15 hours (from alert to fix)

Time period: a 40-hour workweek

15 / 6 = 2.5 hour

MTTR = 2.5 hours for week

10.2. Forecast.

The significant change that is considered to be done in next release is built-in streaming service. The aim is for users to be able to watch films (new modern Estonian cinema), films with Estonian dubbing. This feature includes both free and paid subscription.

System Properties of the Request:

User Story – Requirement	Justification
As a User I want to be able to enable the subscription, so that the user can watch a film in the system	Adding subscription flow Adding payment methods (support Estonian banks, Wise, Revolut, PayPal) Adding streaming possibility

Cost Forecast:

The procurer determines the budget for the request. Within the project's established requirements, the development team may not exceed this amount.

In the forecast session below, the funding for the request is calculated. Salary for the cross-functional team (up to 10 individuals) and infrastructure costs (Cloud Storage and Services) are all included in the budget. A financial strategy has been devised. The procurer creates an overall budget and development plan.

The team and the procurer must agree on a comprehensive development strategy consistent with the overall strategy. This request has a set budget. Budget modifications are frequently prohibited. All stakeholders must follow the project's budget and development plan to guarantee that the request is finished on time and within budget.

Effort Forecast:

The table below estimates the time required to accomplish this request. The team must be approved by the procurer. A project manager, a business analyst, software developers (up to 5 individuals), QA engineers (2 people), and a product owner shall comprise the unit.

Forecast Method:

As a forecast method we decided to use Man-Months (person-months). Because we considered human resources a higher priority. We took phases according to SDLC and described.

We assume that for each phase we allocate time and each specialist in the team describes how much working time they need to spend on a particular phase.

Phase	Time	Product Manager	Business Analyst	Developer	QA Engineer	Product Owner
Requirements & Analysis	1 week / 40 hours	40	40	10	10	40
Planning	1 week	15	15	5	5	5
Design		5	5	10	5	2
Coding & Implementation	8 weeks / 320 hours	40	40	320	320	5
Testing	1 week	40	40	40	40	4
Deployment	1 week	2	2	20	40	2
Maintenance		2	10	20	20	2

A project implemented over 12 weeks (3 month) involves 6 roles (we assume that there is 168 working hours in a month = 21 working days):

- a project manager is working some hours on the project for the entire duration (144 hours) 0.85 month;
- a business analyst is working some hours on the project for the entire duration (152 hours) 0.9 month;
- a developer is working some hours on the project for the entire duration (425 hours) 2.5 month;
- a QA engineer is working some hours on the project for the entire duration (440 hours) 2.6 month;
- a product owner is working some hours on the project for the entire duration (60 hours) 0.35 month

The project effort in person-months would be calculated as follows:

- A project manager – 0.85 person-months;
- A business analyst – 0.9 person-months;
- A developer – 2.5 person-months but we have 5 developers 2.5*5 = 12.5 person-months;
- A QA engineer – 2.6 person-months but we have 2 QA engineers 2.6*2=5.2 person-months;
- A product owner – 0.35 person-months;

Total project effort: 0.85 + 0.9 + 12.5 + 5.2 + 0.35 = 19.8 person-months

A11. Implementing ISKE, ITIL, ISO 9001, or another framework (intro).

11.1. Framework selection.

We decided to implement the ITIL framework because it embraces the latest trends in technologies and service management. It is also the most widely used approach to manage IT services as well as providing a flexible foundation for organizations to integrate digital technology into their business processes. This point is very crucial for us because our main goal is to digitize the service in Estonia. And one more benefit is that new version of ITIL promotes a more agile way of working.

11.2. Overview of implementation.

We implement ITIL 4 approach because it is the most recent edition. Although it includes 34 management practices, we outline below a select number of practices that are relevant to our project:

- **Measurement and reporting**: To support continual improvement of the system, it is necessary to collect data on relevant metrics as well as provide valid assessment of this data. For our system, some of the valid metrics include number of users, films etc.
- **Service configuration management:** The goal of this practice is to ensure that there is access to accurate and dependable information on configuration information and items when it is needed. Although the value of this practice is not direct, it is a foundational item for other practices to work as they should. Therefore, planning for configuration management needs to take place by understanding who needs the information, how it will be used, and the best way for them to obtain it. For our system, we find that after the initial setup of the system, the users will have no need for configuration. Therefore the plan as suggested by the ITIL framework is to make the information available in a

configuration management system (CMS) that will be monitored and maintained by an IT team in charge of the systems.

- **Software development and management:** The purpose of this practice is to ensure that the software meets the requirements of necessary stakeholders, in terms of functionality, reliability, maintainability, compliance, and auditability. Although ITIL 4 does not describe how to develop software, it is added as a practice because it is deemed as an important factor for value creation. Based on previous labs, we find that the software meets all requirements but should also pass security checks, because in this lab we faced with security issues that OWASP revealed, and the software would require some additional development work.
- **Release management:** The purpose of this practice is to plan, schedule and control the movement of releases to the production environment. ITIL defines six approaches for release and deployment management. The one most appropriate for our system is the big branch approach where a new release with instructions for setup is deployed to the server so all clients have access to the new version of the service.
- **Incident Management:** The purpose of this practice is to minimize the negative impact of incidents by restoring normal service operation as quickly as possible. A possible process as described by ITIL involves:

  ○ Distinguishing between service interruptions and service requisitions (like password reset)

  ○ Log all reports as incidents so that their status can be tracked and a historical record maintained

  ○ Escalated to an appropriate support group which in the case of the service will involve development team

  ○ For major incidents, greater urgency needs to be applied.

A12. Retrospective analysis, references, authors

12.1. Retrospective analysis.

Summarizing the result of all 4 Labs, we can note that the whole process helped to analyze and better understand the different parts of software development process such as procurement, development, verification and validation, and maintenance.

During the process we went through all the typical processes of software quality assurance and assessment. We also tried different roles and better understood the responsibilities and expectations for each of the participants and stakeholders of the software development process. During the implementation of testing and verification activities the major part of functional and non-functional requirements was assessed, and acceptance criteria was met successfully.

Analyzing the whole process, we concluded that among things that could be done better in the future projects is ensuring better security of the system. This would require additional work from both procurer and developer roles, as for procurer, the requirements set to such security parts as confidentiality, integrity, non-repudiation, accountability, authenticity.

12.2. Presenting the team

Our team:

Azhar Kazakbaeva – Acquirer (contact person).

Maksym Avramenko (223884IVSM) Developer.

Viktoriia Abakumova (223890IVSM) Tester.

Fillip Molodtsov (223891IVSM) Maintainer.

This division allowed each of the team member provide valuable input specifically on the corresponding stage of software quality assurance and assessment. The requirements provided in the Lab1 by the acquirer were used to develop the main features by the developer in Lab2 and to provide necessary tests by tester in Lab3.

Maintainer’s input in Lab4 was especially important to assess the system’s metrics in production.

12.3. References

For this project the minimal viable product for Estonian filmography website was used. The system and a procurer (Estonian Ministry of culture) were modeled specifically for this course purposes. Among the methodologies that were used in this project are TDD, Risk-based acceptance tests, ITIL.

1. Organization: The Ministry of Culture of Estonia (https://www.kul.ee/en)
2. TDD code: Lab 2 GitLab Repository (https://gitlab.cs.ttu.ee/viabak/idy0204)
3. Test Results: Lab 3 Appendices
4. ITIL https://www.axelos.com/certifications/itil-service-management/