inventi

GUIDEBOOK

Step-by-step Testing Audit

v 1.1

Testing is necessary

SW quality control is often underestimated, especially when it comes to smaller projects. The bigger role the SW solution plays in the business, the more attention is paid to testing. However, paying attention to quality control is right and logical for projects of any scope.

However, paying attention to quality control is right and logical for projects of any scope. **Because of malfunctioning software, we may complicate the path to our goals, load on more work and slow down the development process.** Sooner or later, you will have to correct the errors retrospectively. In such a case, we also have to take into account the risks within business relationships, lost profits or even fines that might arise because of the errors.

Even though testing means more costs, in the long run it protects us from much greater losses that can have a huge impact.

For example, underestimation of testing didn't pay off for the multinational aerospace manufacturer Boeing. The company sent a spacecraft Starliner that didn't have a correctly set internal clock to the International Space Station in December of 2019. The motors didn't start at the right times and poor testing also also affected the control. Thankfully, there was no crew, and it returned to Earth safely. source

One example from the Czech Republic is Tečka which is an application used for managing vaccination certificates. In more instances, the app couldn't start or showed an invalid certificate to its users. source

Does testing have potential?

The software testing market surpassed the global value of 40 million dollars in 2020 and is projected to grow at an annual growth rate of 7 % between 2021 and 2027.

This trend is significantly supported by the growing interest of people in using various apps and the need for companies to go digital. For developers, it only means one thing: testing, testing, testing. In order to keep their customers, companies have to continually update their applications and make them easier to use.

The testing industry is also driven by technological developments in AI and machine learning. When these technologies are used correctly, the testing itself is more flexible and accurate.

How costly is testing?

In the 10 years of INVENTI's operation, we have seen that most medium-sized projects have a development team with roughly this composition:

- **one project manager,**
- **one analyst,**
- **two developers,**
- **two testers.**

The testing budget within the team is therefore approximately 15–20 % of the overall budget.

Radim Dodek,
Delivery Manager, INVENTI

We also have to consider negative factors that could hinder its development. As confirmed by the GMI Software Testing Market 2021–2027 report, these are the most common challenges among companies:

1. Lack of professionals that could handle every step of the process.
2. Prioritizing manual over automated testing because of a greater initial investment.
3. Vague team communication and unclear testing requirements.

Put your testing to a test

In order to find out whether our quality control of the developed product is working efficiently, we have to track several different areas. It's necessary to go through all aspects of the project related to testing.

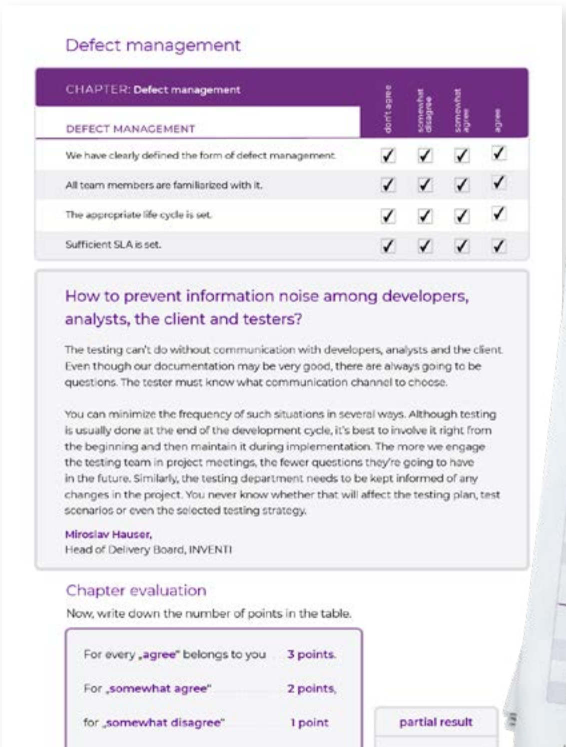
It's not enough to just say that we take care of a specific aspect. **In the course of the audit, it's necessary to focus on the effectiveness of the operation itself.** Control of the team and its communication across all departments is an integral part of this process. It can be a development, business, or project team, DevOps and more.

Testing audit should not be a one-off activity!

It should be a regular cleanup in case there are major changes regarding the project, whether they apply to the staff, project's direction or the processes of its development.

Audit for savings and efficiency – what can it look like in practice?

Thanks to the audit, we can also reveal opportunities where testing can be automated – up to 25% of all tests. This way, we save some capacities of the testing team that gets up to 15% more time. The capacities can be then used in other project areas.



Defect management

CHAPTER: **Defect management** — DEFECT MANAGEMENT	don't agree	somewhat disagree	somewhat agree	agree
We have clearly defined the form of defect management.	✓	✓	✓	✓
All team members are familiarized with it.	✓	✓	✓	✓
The appropriate life cycle is set.	✓	✓	✓	✓
Sufficient SLA is set.	✓	✓	✓	✓

How to prevent information noise among developers, analysts, the client and testers?

The testing can't do without communication with developers, analysts and the client. Even though our documentation may be very good, there are always going to be questions. The tester must know what communication channel to choose.

You can minimize the frequency of such situations in several ways. Although testing is usually done at the end of the development cycle, it's best to involve it right from the beginning and then maintain it during implementation. The more we engage the testing team in project meetings, the fewer questions they're going to have in the future. Similarly, the testing department needs to be kept informed of any changes in the project. You never know whether that will affect the testing plan, test scenarios or even the selected testing strategy.

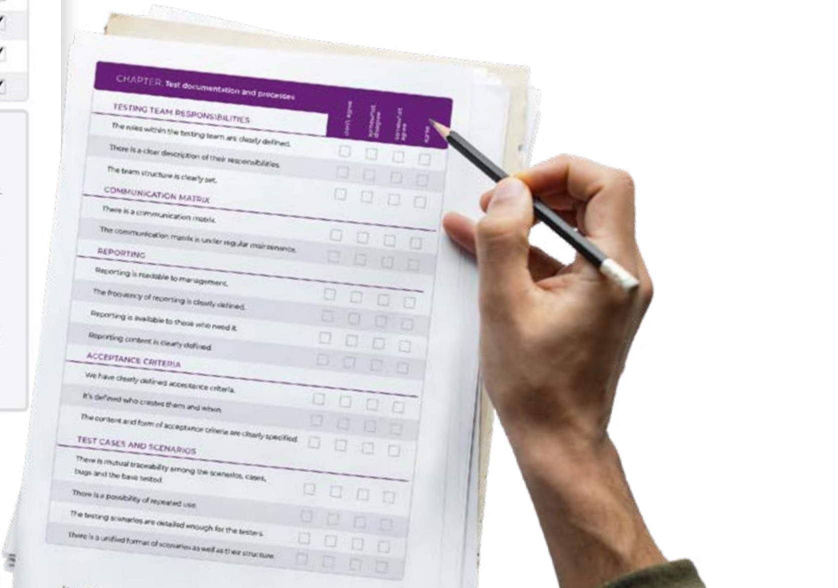
Miroslav Hauser,
Head of Delivery Board, INVENTI

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**" belongs to you	**3 points.**
For „**somewhat agree**"	**2 points,**
for „**somewhat disagree**"	**1 point**

partial result



CHAPTER: **Test documentation and processes**	don't agree	somewhat disagree	somewhat agree	agree
TESTING TEAM RESPONSIBILITIES				
The roles within the testing team are clearly defined.	☐	☐	☐	☐
There is a clear description of their responsibilities.	☐	☐	☐	☐
The team structure is clearly set.	☐	☐	☐	☐
COMMUNICATION MATRIX				
There is a communication matrix.	☐	☐	☐	☐
The communication matrix is under regular maintenance.	☐	☐	☐	☐
REPORTING				
Reporting is readable to management.	☐	☐	☐	☐
The frequency of reporting is clearly defined.	☐	☐	☐	☐
Reporting is available to those who need it.	☐	☐	☐	☐
Reporting content is clearly defined.	☐	☐	☐	☐
ACCEPTANCE CRITERIA				
We have clearly defined acceptance criteria.	☐	☐	☐	☐
It's defined who creates them and when.	☐	☐	☐	☐
The content and form of acceptance criteria are clearly specified.	☐	☐	☐	☐
TEST CASES AND SCENARIOS				
There is mutual traceability among the scenarios, cases, bugs and the base tested.	☐	☐	☐	☐
There is a possibility of repeated use.	☐	☐	☐	☐
The testing scenarios are detailed enough for the testers.	☐	☐	☐	☐
There is a unified format of scenarios as well as their structure.	☐	☐	☐	☐

By yourself or with the help of an external expert?

If you want to do it by yourself, you can't do it without the right team. Such a team should have sufficient experience, motivation and a realistic approach. It is necessary to count with a greater time investment as well.

If you're serious about the audit, get ready for at least 10 days of fulltime work.

When we perform a check like this, we try to get as familiar as possible with the project. We make use of personal meetings, read documentation and testing scenarios and often even become a temporary part of the team.

The result of such an activity is a list of findings and a proposal of several ways to correct or streamline the issue. Then we go through the options with the client and after having a discussion, choose the most suitable way and work it out in detail. The following step is the execution and the evaluation of the benefits, or eventually taking further steps leading to even greater efficiency.

Testing audit on a silver platter

This guidebook may come in handy during your own audit testing execution. You can try to perform an audit like us step by step.

This guide will help you find your way around your testing processes without an external partner. It may calm you down, but also open questions to improve your testing process.

We can't get everything into one guide, including individual needs and specifics of particular projects. However, we have tried our best to cover all the essentials that have proved to be elementary basics during our ten years of experience.

Getting ready for an audit using this guide

If we want to conduct a testing audit, the time devoted to its preparation must be taken into account. As we have mentioned, the key is to choose the right people to do it. They should have an individual view on the testing process and be able to arrange meetings across individual departments, or ask for relevant documents.

You should explain to the testing and project team that it's not your intention to point out their mistakes. The main objective of an audit like this is streamlining the effectiveness in the project delivery process. **If the whole team is not willing to cooperate, the results of the audit will not be accurate.**

Examples from practice: How not to antagonize your team

When we conduct an audit for our clients, we start with defining the objectives and the reason for its execution. Then it's necessary to communicate them to the whole team, so that all members understand them, accept them and get ready for cooperation. That could be a sensitive topic, meaning proper communication is essential. For example, we rely on the following themes in our explanations:

- **Identifying areas the training should be focused on, and increasing the team's know-how.**
- **Investing in testing and looking for new appropriate ways.**
- **Securing more quality inputs from other teams thanks to well-set testing.**
- **Purchasing new hardware or software and setting up the process of its use.**
- **Freeing up the capacities of the testing team to start dealing with areas they draw attention to in the long term.**

Jan Kaštura
Chief Sales Officer & Co-founder, INVENTI

Reserve one day to prepare the audit and roughly five days for its execution.

What do you need to do in advance?

1. Prepare the documentation related to the project you are auditing.
   - Business documentation
   - Technical documentation
   - Test documentation
   - Project plan

2. Prepare a list of tools you use to work on the project and testing.
   - Project management
   - Test management
   - Defect management
   - Testing tools

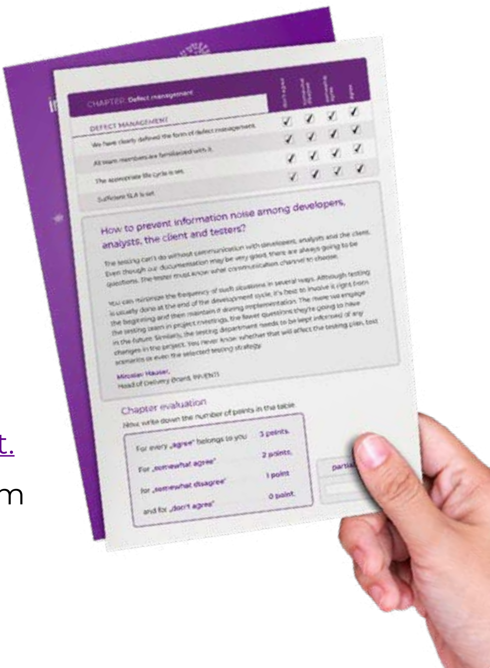
3. Schedule meetings with departments and managers.
   - Release management
   - Project management
   - Test management
   - Analytical team
   - Support (L1,L2,L3)

4. If possible, ask for feedback from the production environment.

We recommend you to create a single document where you'll take all the notes during the audit in progress. Keep coming back to them and put the individual "pieces" together into a complete mosaic.

In order to perform a basic testing audit, go through the individual points in this guide and take notes about how much you agree with them. There's going to be an evaluation for each phase where you'll get an idea about whether you're doing good or whether there's room for improvement.

The whole checklist can be printed out and used as a worksheet. Answer truthfully! This is the only way to gain maximum knowledge.



CHAPTER: Defect management

DEFECT MANAGEMENT	[illegible]	[illegible]	[illegible]	[illegible]
We have clearly defined the form of defect management.	✓	✓	✓	✓
All team members are familiarized with it.	✓	✓	✓	✓
The appropriate life cycle is set.	✓	✓	✓	✓
Sufficient SLA is set.	✓	✓	✓	✓

How to prevent information noise among developers, analysts, the client and testers?

The testing can't do without communication with developers, analysts and the client. Even though our documentation may be very good, there are always going to be questions. The tester must know what communication channel to choose.

You can minimize the frequency of such situations in several ways. Although testing is usually done at the end of the development cycle, it's best to involve it right from the beginning and then maintain it during implementation. The more we engage the testing team in project meetings, the fewer questions they're going to have in the future. Similarly, the testing department needs to be kept informed of any changes in the project. You never know whether that will affect the testing plan, test scenarios or even the selected testing strategy.

Miroslav Hauser,
Head of Delivery Board, [illegible]

Chapter evaluation

Now, write down the number of points in the table

For every „agree" belongs to you	3 points
For „somewhat agree"	2 points
for „somewhat disagree"	1 point
and for „don't agree"	0 point

partia

Audit execution

Introduction to the project

Testers should know the product to the smallest of detail. They daily communicate with developers, analytics, and business members of the team. In order to propose appropriate test scenarios, set the right priority for them and plan them in time, they must know what the product is as well as how and why we make it.

They should be involved in the project right from the beginning. If they're not, they should have access to the project documentation or ideally go through an introductory workshop.

Explaining who we're developing the product for, as well as its purpose and objective, also plays an integral role in correct and effective testers' engagement. What the end user should behave like or the devices they should use. Non-functional requirements, meaning requirements for security, or the number of users that can use the software or hardware at once, also belong to crucial information.

Testers are therefore able to design an appropriate testing strategy, as well as to plan and prepare the relevant test data. **At this point, we often discover the first shortcomings already.**

CHAPTER: **Introduction to the project**

BUSINESS REQUIREMENTS	don't agree	somewhat disagree	somewhat agree	agree
We have clearly defined our business requirements.	☐	☐	☐	☐
We have a detailed description of business requirements.	☐	☐	☐	☐
The testing team clearly understands these business requirements.	☐	☐	☐	☐
END USERS				
We have clearly defined the end user of the tested application.	☐	☐	☐	☐
We have clearly defined use cases.	☐	☐	☐	☐
The testing team is properly familiarized with these definitions.	☐	☐	☐	☐
END DEVICES				
We have a list of the used end devices.	☐	☐	☐	☐
We know which end devices are priority (ideally based on the available data).	☐	☐	☐	☐
The testing team has priority end devices at hand and is able to work with them.	☐	☐	☐	☐

Examples from practice:
Diversity of end devices

The end device is not always a phone, a computer or a tablet. For example, on our significant project for a manufacturer of recreational vehicles, we work on different parts of the caravan. It's necessary to perform testing with variable control panels and sometimes right in the caravans.

Miroslav Novotný

Test Manager, INVENTI

continue to the next page

CHAPTER: **Getting to know the project**

NON-FUNCTIONAL REQUIREMENTS	don't agree	somewhat disagree	somewhat agree	agree
We have an updated list of non-functional requirements.	☐	☐	☐	☐
We have clearly defined specific non-functional requirements.	☐	☐	☐	☐
DOCUMENTATION				
We have clear and complete documentation of the product/project.	☐	☐	☐	☐
The testing team has this documentation at their disposal.	☐	☐	☐	☐
Documentation is versed.	☐	☐	☐	☐
ARCHITECTURE				
We have a clear description of the architecture and its sub-parts.	☐	☐	☐	☐
The testing team is familiarized with the architecture.	☐	☐	☐	☐

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**“ belongs to you	**3 points**.
For „**somewhat agree**“	**2 points**,
for „**somewhat disagree**“	**1 point**
and for „**don‘t agree**“	**0 point**.

partial result

Test documentation and processes

The testing strategy is often considered to be useless by management, but sometimes even by the testers themselves

That's a mistake.

Can you imagine building a house without a design? Without a thorough project plan and documentation? Have you ever tried to drive to a place you've never been before without a map?

The same can be applied to testing strategy, plan and scenarios as well. The objective of the strategy is, among others, to describe the way of testing the product, how many people are needed, what tools we are going to use and what type of tests we are going to pay attention to. Everyone who has anything to do with testing must familiarize themselves with this document. The goal is for the entire team to have the same approach to product testing and for each member to know what to do. The test strategy is to clarify the division of responsibilities during testing.

The following important document is a testing plan. There, we describe what, how and when we are going to test. We work with the testing plan during the process and modify it based on facts and needs that might arise.

What might a test strategy look like?

Want to see how we approach test strategies at INVENTI? Take a look at a simplified **test strategy template**.

This is always a starting point for us to create an individualized and detailed strategy that we use on both internal and external projects.

Test strategy

Before moving on, make sure that you and your team are clear on the strategy.

CHAPTER: **Test documentation and processes**

TEST PLAN	don't agree	somewhat disagree	somewhat agree	agree
We have defined a comprehensive test plan and determined who creates it.	☐	☐	☐	☐
We make workload estimates for testing.	☐	☐	☐	☐
We define workload estimates by the in scope, out of scope method.	☐	☐	☐	☐
We define test data for every run.	☐	☐	☐	☐
We have determined who defines and creates test data, including the process of its definition.	☐	☐	☐	☐
The test estimates are reflected in the project plan.	☐	☐	☐	☐

Test plan in 10 minutes

Are you interested in how to write the test plan quickly and easily? Take a look at the manual of a recognized expert James Whittaker, professor at the Florida Institute of Technology and test director at Google. You can find it on the Google Testing blog, or on the record from EuroSTAR Software Testing Conference.

CHAPTER: **Test documentation and processes**

TESTING TEAM RESPONSIBILITIES	don't agree	somewhat disagree	somewhat agree	agree
The roles within the testing team are clearly defined.	☐	☐	☐	☐
There is a clear description of their responsibilities.	☐	☐	☐	☐
The team structure is clearly set.	☐	☐	☐	☐

continue to the next page

CHAPTER: **Test documentation and processes**

COMMUNICATION MATRIX	don't agree	somewhat disagree	somewhat agree	agree
There is a communication matrix.	☐	☐	☐	☐
The communication matrix is under regular maintenance.	☐	☐	☐	☐

Reporting

During testing, you need to keep track of the testing activities. We have to know whether our efforts are bearing fruit. Similarly, we should have an overview of the errors found and their lifecycle.

That's what regular reporting is for.

The most effective option is to have real-time reporting. Examples are various dashboards in Jira or Confluence

During audits, we often find out that testing takes place randomly. We hardly ever get answers to questions like:

- How was the testing conducted?
- What data was used?
- Who did the test?
- When was the test conducted?

Reporting template

We base reporting of every project on our standardized reporting template. Of course, there must always be clear specifications and individualization according to the specific needs and strategy of the project. The attached pattern can still serve as a good source of inspiration anyway.

continue to the next page

CHAPTER: **Test documentation and processes**

REPORTING	don't agree	somewhat disagree	somewhat agree	agree
Reporting is readable to management.	☐	☐	☐	☐
The frequency of reporting is clearly defined.	☐	☐	☐	☐
Reporting is available to those who need it.	☐	☐	☐	☐
Reporting content is clearly defined.	☐	☐	☐	☐

For optimal results, these situations must be prevented.

If we have a Story / Task / Requirement – Test scenario – Execution test link set up, we can look back at it any time to see why a specific error fell through to production. Thanks to that, we learn a lesson and don't repeat it.

It's important not to neglect the maintenance and writing of testing scenarios. Similarly, we need to record their execution.

In order to be able to create testing scenarios, we must have proper documentation secured. Besides business and IT documentation, acceptance criteria are very important as well. This way, the test analyst knows what needs to be done.

continue to the next page

CHAPTER: **Test documentation and processes**

ACCEPTANCE CRITERIA	don't agree	somewhat disagree	somewhat agree	agree
We have clearly defined acceptance criteria.	☐	☐	☐	☐
It's defined who creates them and when.	☐	☐	☐	☐
The content and form of acceptance criteria are clearly specified.	☐	☐	☐	☐

What might acceptance criteria look like in practice?

Acceptance criteria define the expected behavior of the application and partial features from the user's perspective, under which the application can be accepted as functional. Let's take an example from a specific e-shop to whom we provide testing services. In this e-shop, the customer can choose a purchase reward. What acceptance criteria did we define when developing a feature like this?

- The customer gets to choose out of three products.
- Each product will have its price crossed out.
- The customer can choose either one product or none.

CHAPTER: **Test documentation and processes**

TEST CASES AND SCENARIOS	don't agree	somewhat disagree	somewhat agree	agree
There is mutual traceability among the scenarios, cases, bugs and the base tested.	☐	☐	☐	☐
There is a possibility of repeated use.	☐	☐	☐	☐
The testing scenarios are detailed enough for the testers.	☐	☐	☐	☐
There is a unified format of scenarios as well as their structure.	☐	☐	☐	☐

continue to the next page

CHAPTER: **Test documentation and processes**

TEST CASES AND SCENARIOS	don't agree	somewhat disagree	somewhat agree	agree
There is a validation process set.	☐	☐	☐	☐
The test scenario is included in test planning.	☐	☐	☐	☐
Within the scenarios, prioritization is used.	☐	☐	☐	☐
Scenarios are created and assigned to every functional or non-functional requirement.	☐	☐	☐	☐
There are methods for test analysis.	☐	☐	☐	☐

TEST EXECUTION				
Both the progress and the results of the test are recorded.	☐	☐	☐	☐
Traceability is secured.	☐	☐	☐	☐
There is an ongoing retrospective evaluation.	☐	☐	☐	☐
The team has the capacity to handle the tests.	☐	☐	☐	☐
The team has sufficient knowledge and skills for execution.	☐	☐	☐	☐

How to prevent skeletons in the closet?

During integration testing of a complex IT solution, we naturally find errors. However, the errors may not be dealt with at all, take unnecessarily long to resolve, or the team may not be able to agree on how to resolve them. We have found out that it can be avoided by a clear approach and everything starts with the error input itself. It's more than appropriate to set a unified form for entering an error. It's necessary to come to an agreement about it, especially with developers and analysts so all needed and available information is entered. The following step is to clarify the priority and severity of errors and the expected time to resolve them.

Zdeněk Vencl
Project Manager, INVENTI

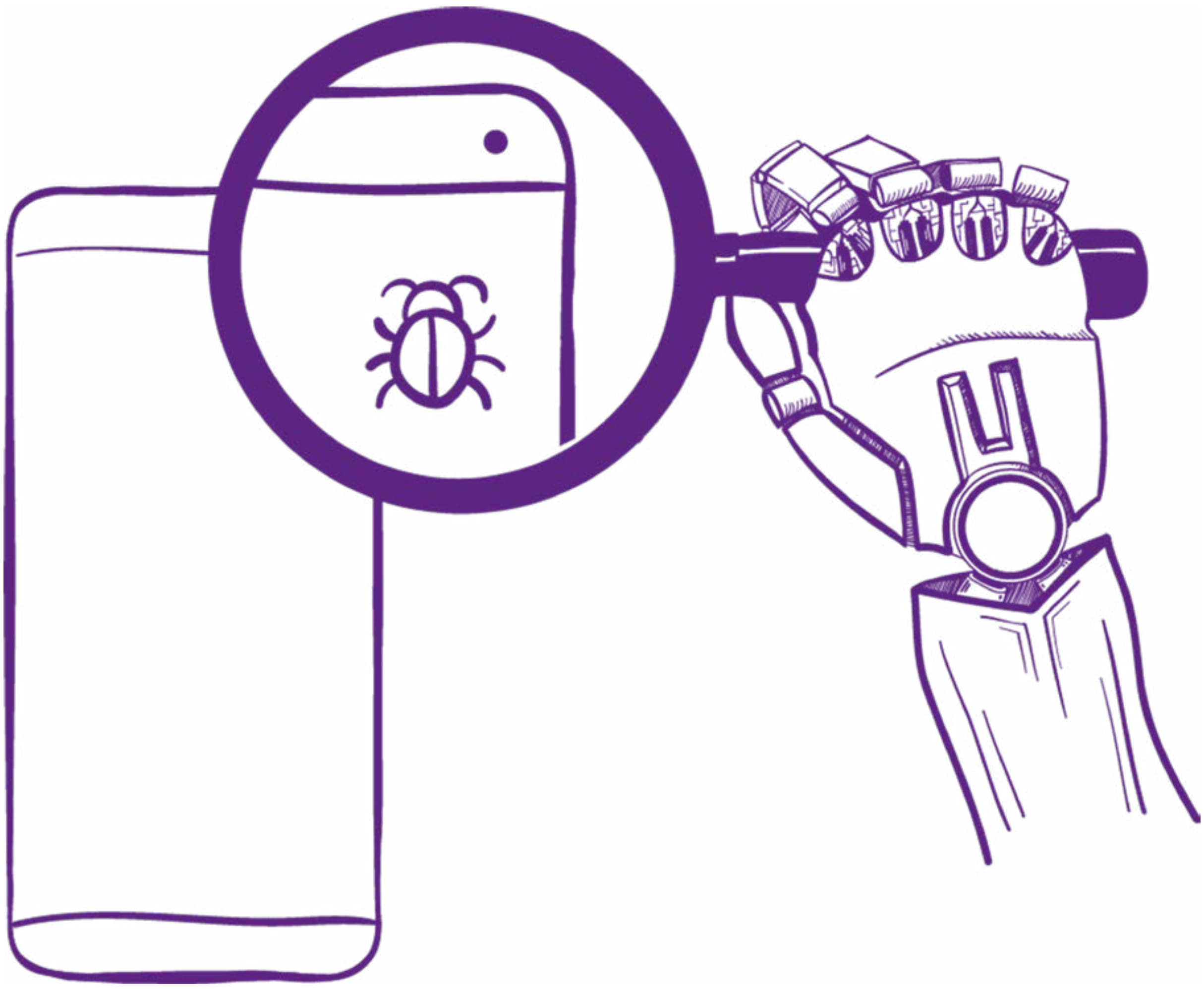
continue to evaluate the chapter

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**“ belongs to you	**3 points**.
For „**somewhat agree**“	**2 points**,
for „**somewhat disagree**“	**1 point**
and for „**don‘t agree**“	**0 point**.

partial result



Defect management

CHAPTER: **Defect management**	don't agree	somewhat disagree	somewhat agree	agree
DEFECT MANAGEMENT				
We have clearly defined the form of defect management.	☐	☐	☐	☐
All team members are familiarized with it.	☐	☐	☐	☐
The appropriate life cycle is set.	☐	☐	☐	☐
Sufficient SLA is set.	☐	☐	☐	☐

How to prevent information noise among developers, analysts, the client and testers?

The testing can't do without communication with developers, analysts and the client. Even though our documentation may be very good, there are always going to be questions. The tester must know what communication channel to choose.

You can minimize the frequency of such situations in several ways. Although testing is usually done at the end of the development cycle, it's best to involve it right from the beginning and then maintain it during implementation. The more we engage the testing team in project meetings, the fewer questions they're going to have in the future. Similarly, the testing department needs to be kept informed of any changes in the project. You never know whether that will affect the testing plan, test scenarios or even the selected testing strategy.

Miroslav Hauser,
Head of Delivery Board, INVENTI

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**" belongs to you . . . **3 points**.

For „**somewhat agree**"................... **2 points**,

for „**somewhat disagree**"............... **1 point**

and for „**don't agree**"..................... **0 point**.

partial result

Communication

CHAPTER: **Communication**				
COMMUNICATION	don't agree	somewhat disagree	somewhat agree	agree
The testing team has the opportunity to communicate with the business and development teams.	☐	☐	☐	☐
The testing team is involved in all project meetings.	☐	☐	☐	☐
The testing team is a part of the analytical phase.	☐	☐	☐	☐
Third parties answer questions of the testing team on time.	☐	☐	☐	☐
The testing team gets all important information regularly and on time (i.e. release, changes in the application from other teams, etc.).	☐	☐	☐	☐

Don't waste your time

We often find that a lot of time is wasted on delays and tutoring for missing knowledge. If possible, we recommend separating the testing environment from the development one – testers are therefore able to test regardless of the development. From time to time, you need to deploy a new version for testing, of course. In this case, we recommend ensuring the testing team is sufficiently informed about the planned deployment and not deploying right before the planned release to production.

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**“ belongs to you	**3 points**.
For „**somewhat agree**“	**2 points**,
for „**somewhat disagree**“	**1 point**
and for „**don‘t agree**“	**0 point**.

partial result

Test environment

CHAPTER: **Test environment**

TEST ENVIRONMENT	don't agree	somewhat disagree	somewhat agree	agree
The test environment is stable and available.	☐	☐	☐	☐
Deployment windows are clearly defined.	☐	☐	☐	☐
The testing is updated on the unavailability of the environment on time.	☐	☐	☐	☐
A development completion date is clearly defined, after which the code cannot be freely edited.	☐	☐	☐	☐
The environment corresponds with the testing needs and pre-production.	☐	☐	☐	☐
Maximum support for the test environment is secured.	☐	☐	☐	☐

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**“ belongs to you	**3 points**.
For „**somewhat agree**“	**2 points**,
for „**somewhat disagree**“	**1 point**
and for „**don‘t agree**“	**0 point**.

partial result

continue to the next page

Team knowledge

The team knowledge comes from a pre-set testing strategy. There should always be one senior tester, or even better a test lead, in the testing team. It's not needed that they control all the selected tools, but that they know where to get information to lead junior colleagues. They should also have good communication skills.

CHAPTER: **Team knowledge**

TEAM KNOWLEDGE	don't agree	somewhat disagree	somewhat agree	agree
The team has good knowledge of the selected methodology.	☐	☐	☐	☐
Sufficient experience in the team is ensured.	☐	☐	☐	☐
The team has good knowledge of the selected tools and technologies.	☐	☐	☐	☐
Continuous training of the team members is secured.	☐	☐	☐	☐
The team has sufficient knowledge of the given business topic.	☐	☐	☐	☐
Knowledge of test management is ensured.	☐	☐	☐	☐

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**“ belongs to you . . . **3 points**.

For „**somewhat agree**“.................. **2 points**,

for „**somewhat disagree**“............... **1 point**

and for „**don‘t agree**“ .... **0 point**.

partial result

What about the testing team setup and savings in automated testing?

There is usually one tester in teams of 3–4 developers. In case we involve automation, we need to count with 1.5–2 testers for the same number, at least in the first phases of the project. Automation saves a lot of time, but there's a greater investment needed in the beginning. Even test maintenance in continuous development requires a certain capacity.

If you're building a simple e-shop where the development team consists of 1 Senior Technical Lead, 1 Backend Developer and 1 Frontend Developer, your team could look like this:

1 × Test Lead - responsibility for test management, test analysis and execution for manual testing.

1 × Test Analyst - responsibility for manual testing, analysis and selection of tests for automation.

The role of Test Lead can be shared with another role, such as Product Owner, Project Manager or Test Analyst. Test Lead is primarily responsible for test management on the project. At the same time, they can help with test analysis and test execution. On smaller projects, this will take roughly 0.25 – 0.5 FTE.

Filip Kadlec,
Head of Testing, INVENTI

Tools

CHAPTER: **Tools**				
TOOLS	don't agree	somewhat disagree	somewhat agree	agree
The team has the knowledge needed and has quality tools for test management, defect management and test documentation.	☐	☐	☐	☐
The team has tools for all selected testing levels at their disposal.	☐	☐	☐	☐
The tools are in alignment with the selected development technologies.	☐	☐	☐	☐

Proven testing tools

Test administration in JIRA

Do you use JIRA? Then choosing one of the possible test management add-ons is the best solution. Zephyr, XRay, or Test rail, are some of them. If it's not worth it because of licenses, you can take a look at tools like Spira or TestLink. You can find a detailed overview here.

The most widely used tools and frameworks.

Most testers use tools like Postman for API testing, another option is SoapUI. For automation, they use Cypress, Java Selenium/Selenaid and RobotFramework is widely used as well.

Chapter evaluation

Now, write down the number of points in the table.

For every „**agree**" belongs to you	**3 points**.
For „**somewhat agree**"	**2 points**,
for „**somewhat disagree**"	**1 point**
and for „**don't agree**"	**0 point**.

partial result

continue to evaluate the test

Audit evaluation and following steps

In the table below, fill out the total points from individual chapters and then add them up. Below that, we're going to list what you need to work on based on your result.

OVERALL EVALUATION	POINTS
Familiarization with the project	
Test documentation and processes	
Defect Management	
Communication	
Testing environment	
Team knowledge	
Tools	
In total	

Less than 25 points

It is obvious from the results that your testing processes are not set up sufficiently. You are likely to experience significant errors falling through to production. Another consequence may be chaotic rather than systematic correction of errors. Also, companies with a score this low usually don't have an overview of the test coverage and what specific features are covered by the testing scenarios. Besides, they're missing the test execution result.

In this case, we recommend focusing on the audit thoroughly and analyzing the weakest points to start working on. We can take a look at your results of the audit, identify the weakest points and suggest the next steps. An example of these steps may be setting and implementing a correct test strategy, training your internal team for best practices, and eventually building an appropriate testing team.

I WANT A FREE CONSULTATION

26–50 points

You are well aware of the need to test within your development. Even though it may seem that this result is good enough, it doesn't mean that you don't have anything to work on. Some of your features might not be tested enough, the testing team might not deliver on time or the testers may keep the rest of the team busy too often. The causes may involve flaws in the testing strategy, insufficient knowledge of best practices, unclearly defined roles in the team, or occasional "omission" of testing processes.

You should take a closer look at the results of the audit, identify the weak areas and think about their impact on the project and business. If you need help specifying potential impacts, we would love to help you identify and eliminate them.

I WANT A FREE CONSULTATION

50 and more points

Amazing! Testing is an essential part of your processes, which is great. Don't rest on your laurels, though! There is usually room for sophisticated automation, reporting or streamlining flow across SDLC. Testing is continuously being developed, and new tools and technologies are being created.

Would you like to take your testing to the next level? Don't hesitate to contact us! We can analyze your situation and suggest specific steps to make on the path to a brilliantly functioning project.

I WANT A FREE CONSULTATION

An audit is not the end

Jakmile máme jasno v tom, jak aktuálně náš proces testování vypadá, je potřeba si říci, co dál. V tomto bodě doporučujeme vyhotovit analýzu rizik. Cílem je vybrat nejdříve body, které přináší největší riziko pro společnost. Ty rozpracujeme jako první.

How to analyze risks?

Articles from **Guru 99**, **Softwaretestinghelp.com** or **PMI.org** will help you with risk analysis.

Even though the results can tell you a lot, the situation always needs to be assessed from various points of view. Then it's necessary to approach every project individually. For example, the importance of software in your business is a crucial factor in your performance. If it's not that high, a bad result doesn't have to be such a disaster. On the other hand, if the software plays a crucial role in your business, even a slight deviation from the perfect condition might mean great risk and damage.

In the next step, start working on a realistic plan to quickly and simply implement changes. There may be many factors coming into play, including license prices, recruitment of new people, provision of external training or learning new methodology. Points like the addition of acceptance criteria may have a quick solution while using the existing resources.

Next, creating a work schedule is needed. It's really important to set out specific steps, who's going to make them, when they are going to be completed, and eventually when we are going to check them. Don't forget retrospective assessment of the change implementation also.

Keep in mind that the result is informative only. If you want to dive deeper, don't hesitate to contact us. We offer you a free consultation where we go through the results together, realistically evaluate and set the steps needed for further growth.

I WANT A FREE CONSULTATION

Useful links

Want to know more about the topic?
Check out the following links:

How Google Tests Software – James A. Whittaker

Top 10+ Best Software Testing Books (Manual And Automation Testing Books) – Software Testing Help

ISTQB® Effectiveness survey 2019-2020 – ISTQB

Software Testing Market 2021-2027 - Global Market Insights

Google Testing Blog

How To Write Test Strategy Document (With Sample Test Strategy Template) – Software Testing Help

Risk Based Testing: Approach, Matrix, Process & Examples – Guru99

The Ultimate Guide To Risk Based Testing: Risk Management In Software Testing – Software Testing Help