



**the test
leaders**
we become you



Risk

the test leaders

Mechelsesteenweg 277
B-1800 Vilvoorde

Web

www.thetestleaders.com

Tel

+32 (0)2 252 12 17

Fax

+32 0(2) 252 12 18

eBook

Table of Content

1.	Foreword	3
2.	When we perform a patch or maintenance activities, we may introduce new issues. How can we avoid this?.....	4
2.1	Setting the scene	4
2.2	Our answer	4
2.3	In short	8

1. Foreword

The FAQ

To be successful a test leader needs to master skills within 10 different knowledge areas. By referring to our highly visual testing intelligence™ framework, you can go:

“From theory to reality in 30 minutes”

Because we believe the core of software testing is about managing risks, we started gathering our experience and answering frequently heard questions in the RISK MANAGEMENT knowledge area.

The questions will tackle different aspects of risk management in context of projects and operational situations.

2. When we perform a patch or maintenance activities, we may introduce new issues. How can we avoid this?

2.1 Setting the scene

What is a patch? A patch is a piece of software designed to fix problems with, or update a computer program or its supporting data. This includes fixing security vulnerabilities and improvements to usability or performance. Though meant to fix problems, poorly designed patches can sometimes introduce new problems!

Patches or maintenance activities take place in a controlled environment running in production, supported by operational teams following certain processes.

The general mistake made is that a patch is not treated as a small “project”. Patches are often taken care of within one team. Therefore, it often will not benefit from the input or review from other stakeholders or from any project management or test process.

Of course, when managing patches a lighter, more pragmatic process must be used, but nevertheless based on the same methodologies and best practices used for projects.

2.2 Our answer

You must accept that a patch is of the same nature than any other project and thus should follow the same type of treatment although lighter as the scope is by nature smaller than a project.

“It is not the effort of the change but its impact and the companies appetite for risk which will determine the actions needed to be undertaken.”

By simplifying the processes and deliverables of a project to fit the “lighter” nature of a patch, you benefit from the added value of the tools and processes of a project but do not overkill it with an oversized administrative burden.

This approach has also as an objective to educate all stakeholders involved in deploying patches and performing maintenance activities. The development of a lightweight process indicates that this type of changes **MUST** be delivered in production in a controlled way.

The patch or maintenance process is triggered by a request for a patch or maintenance activity.

To perform a controlled release of patches, our 5-step Risk Approach can be used in preparation of the release:

1. Identify Risks
2. Analyse Risks
3. Define Risk Appetite
4. Define Test Strategy
5. Manage Risks

2.2.1 Step 1: Identify Risks

To perform Risk Identification, you have to consider at least the following areas:

1. Criticality of the system(s) impacted by your patch

Although lighter than a project, you will probably not treat an update regarding your intranet the same way you treat an update regarding your accounting or security systems. Not even mentioning safety critical systems.

2. Scope of the change

The exact scope of the change needs to be decided upfront (avoid opportunity updates) and formalized to ensure that the risk analysis that will be later applied takes into account the right elements.

3. Complexity of the change

We all know that the higher the complexity the higher the risk that something goes wrong while performing the change, so complexity will influence the implementation approach.

2.2.2 Step 2: Risk Analysis

Once the risk identification has been done, you can start to perform your Risk Analysis. You will know if the system you are going to touch is critical or not for your internal organisation or for your customers, you will know the different components (software and hardware) you will interact with and whether the change is complex or not (overall and per component). All of these are critical Inputs to perform a proper risk analysis.

2.2.3 Step 3: Determine Risk Appetite

Identifying risks and analyzing them to get a good understanding of the changes you are making is one thing. Once you know the risks and their severity you need to decide which residual risk is acceptable. Meaning, you will have to decide to what level you need to reduce the risks before you can implement the patch.

2.2.4 Step 4: Define Testing Strategy

Based on the previous 3 steps, you can define a Risk Based Testing Strategy:

Depending on the risk and the risk appetite more or less formality will be needed and different test techniques and test depth are required.

- A risky change will need full-blown and formal testing
- A change with little risk can be tested lighter and less formal.

Using a Risk Based Test Strategy approach, you will know which deliverables and tools to use (or not) to manage properly your initiative without an oversized administrative burden but still taking benefit from the best practices from project, risk and test management.

Follow the patch processes with the level of formalization recommended in the previous step.

2.2.5 Step 5: Manage Risks

Of course do not forget that although the formalization will be lighter, the risk management will still need to be maintained during the entire patch lifecycle to ensure that quality is delivered within the agreed level and risks are within acceptable boundaries.

2.3 In short

Accept that a patch is a lightweight project!

By using our 5-step Risk Approach:

1. Identify Risks
2. Analyse Risks
3. Determine Risk Appetite
4. Define Test Strategy
5. Manage Risks

You will:

1. Protect your production environment from unwanted and unforeseen issues and regressions
2. Formalize maintenance activities
3. Help to educate operational teams by getting them acquainted with project management processes (which could be useful when full grown projects or programs are assigned to them)
4. Educate requestors by formalizing the scope, risk level and stakeholders of patches (so they can ask themselves "Is this really necessary? Do I really need to do this now?")