



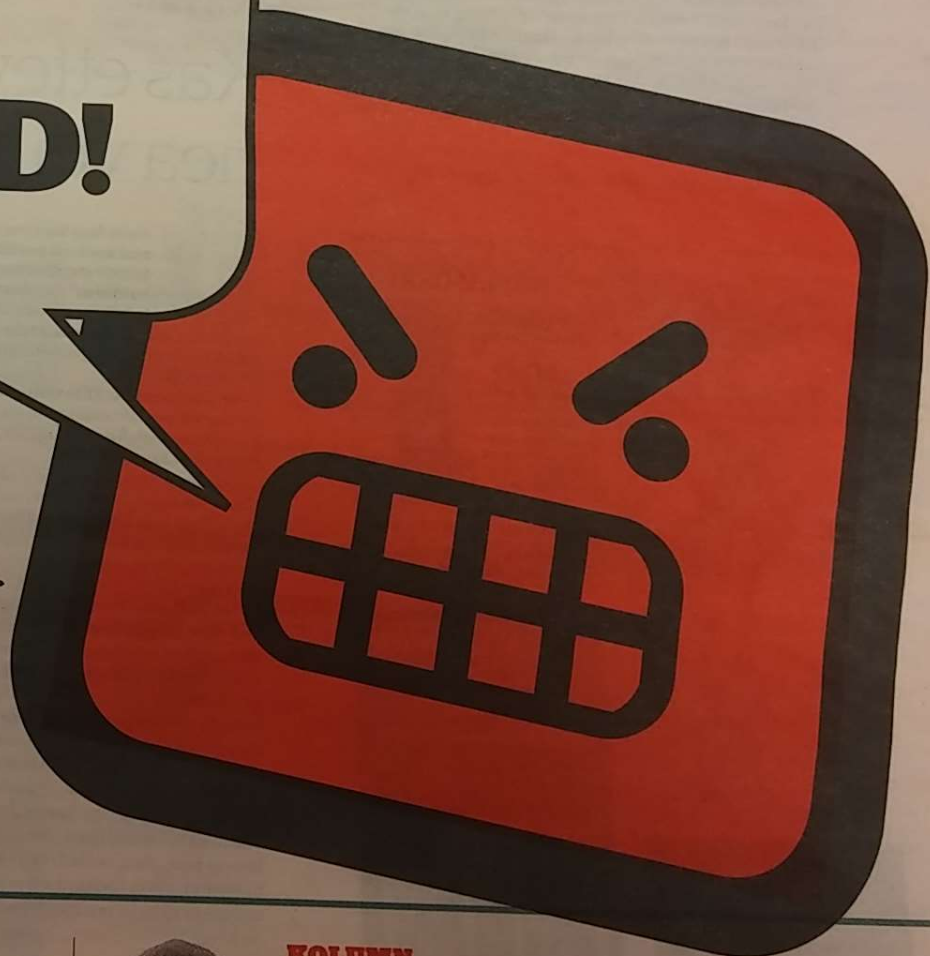
Worst Practices in Software Quality a.k.a How to Deal with Risks

Maili Markvardt

14.11.2019

OLEME VÄGA KURJAD!

► Tehnilised tõrked maksuameti veebikeskkonnas ei lase raamatupidajatel tähtajaks esitada tulu- ja sotsiaalmaksu deklaratsioone. ▶4-5



We don't make software

- We help developers and customers in creating better software
- We prefer to be hands-on in development process and side-by-side with business
- Our area of expertise includes all types of testing, manual as well as automated, and supporting activities like test management and training
- In a way, we help mitigating software related risks

**SOFTWARE IS ALWAYS
A SOLUTION TO A PROBLEM
NOT A „THING“ IN ITSELF**

1 Andmete sisestamine 2 Andmete kinnitamine 3 Tulemus

Teie tehing läks katki

https://public.pensionikeskus.ee/public/commonInfo.do#/main

Search



ILI MARKVARDT
to:

Minu pensionikonto koondväljavõte



Imed

Idused

ingud

amaksud

tagasiside ankeet

nimekiri

su tõend

III samm avatud:
03.08.2018

Pensionikonto väliavõte

Pensionikonto väärtus :EUR

[Vaata konto saldot](#)

[Vaata konto väljavõtet](#)

Soovin järgmisel aastal saada pensionikonto teatist oma e-posti aadressile:

KINNITA



Different meaning for different people and roles

No universal definition of RISK exists

Three ingredients to make a „risk“

- However, 3 characteristics are common in all definitions
 - The potential loss must exist
 - Uncertainty with respect to the eventual outcome must be present
 - Some choice or decision is required to deal with both – uncertainty and potential loss



In short ...

- Risk is the possibility of suffering loss*
 - You won't like it happening
 - It's not certainty
 - You may avoid or soften it if you invest into

- *[Dorofee 1996]



Big risk? Small risk?: Measuring risk

- Probability – how likely it is to happen
- Impact – how hard it hits/measure of loss
- Risk Exposure = Probability * Impact



Examples – probability and impact

- Always agree on probability and impact measure scale
 - Low – Medium – High?
 - Trivial - Minor - Medium – Major – Critical – Blocker?
- Whatever scale units you use, units must be unambiguously defined
 - Low – internal issues/inconveniences/inefficient time usage
 - Medium – impacts up to 100 domestic (not business) customers
 - High – financial loss or negative correspondence in media
- Scales depend highly on application domain (e-commerce vs medical systems)



Risk management

- Systematic approach for minimizing exposure to potential losses
- It provides supporting framework for all activities starting from
 - determining risks
 - Prioritizing
 - dealing with risks



**REQUIREMENT
=
VALUE TO SOME
STAKEHOLDER**

1 Andmete sisestamine 2 Andmete kinnitamine 3 Tulemus

Teie tehing läks katki

https://public.pensionikeskus.ee/public/commonInfo.do#/main

Search



ILI MARKVARDT
to:

Minu pensionikonto koondväljavõte



Imed

Idused

ingud

amaksud

tagasiside ankeet

nimekiri

su tõend

III samm avatud:
03.08.2018

Pensionikonto väliavõte

Pensionikonto väärtus :EUR

[Vaata konto saldot](#)

[Vaata konto väljavõtet](#)

Soovin järgmisel aastal saada pensionikonto teatist oma e-posti aadressile:

KINNITA

Risk & Requirement – a perfect couple

- Requirements and risks always go together
 - If there's a requirement, something can go wrong (risk)
 - If we are afraid of something going wrong, there's some value (requirement)
- This coupling can assist you in determining and checking requirements' priorities and risks' impact
 - These must be coherent: important requirement = significant impact
- Hint for self-check
 - Double-check that couples exist



Let's see example: e-service performance (1)

- Req: must respond in X seconds with Y concurrent users
- Business risk: due to slow response times, e-service is not usable and customer service is overloaded. But our aim is to reduce the number CC employees.
- Tech (IT) risk: existing servers may not perform as expected and we may have to re-write the code or buy additional computing power.

How risks affect software and services (development)

- All risks have effect to software and service quality
 - Customer and user satisfaction
 - Efficiency of resource utilisation
 - Preventing problems instead of fire-fighting
 - Ability to operate in crisis

What do we mean by „absence quality“?

- Outcome is faulty
 - Functionality
 - Non-functional characteristics (usability, security, performance)
- Outcome does not meet users' nor op's' expectations
- Documentation is not sufficient
- Difficulties in maintenance and changes

- Project is behind schedule
- Project exceeds budget

- Relationships are ruined

Root causes: idea phase

- Problem to be solved is unclear
- Bad procurement
 - From day 1 we are already behind schedule
- Too ambitious shedule
- ROI not analysed

Root causes: Initiation

- Lack of knowledge in application domain
 - Important terms, basic business processes
 - Risks – what is important in this system?
- Bad requirements, lack of „analysis“
 - Underestimating the importance of understanding the domain and what we are doing
 - Typical mistakes in RE (unclear, ambiguous requirements)
 - Person who is supposed to do analysis is not up to the job (yet)
- Underestimating risks related to team
 - Competence, especially non-it members of the team
- Stupid mistakes in the project timeline planning
 - Forgetting public holidays and holiday seasons
- Improper development methodology

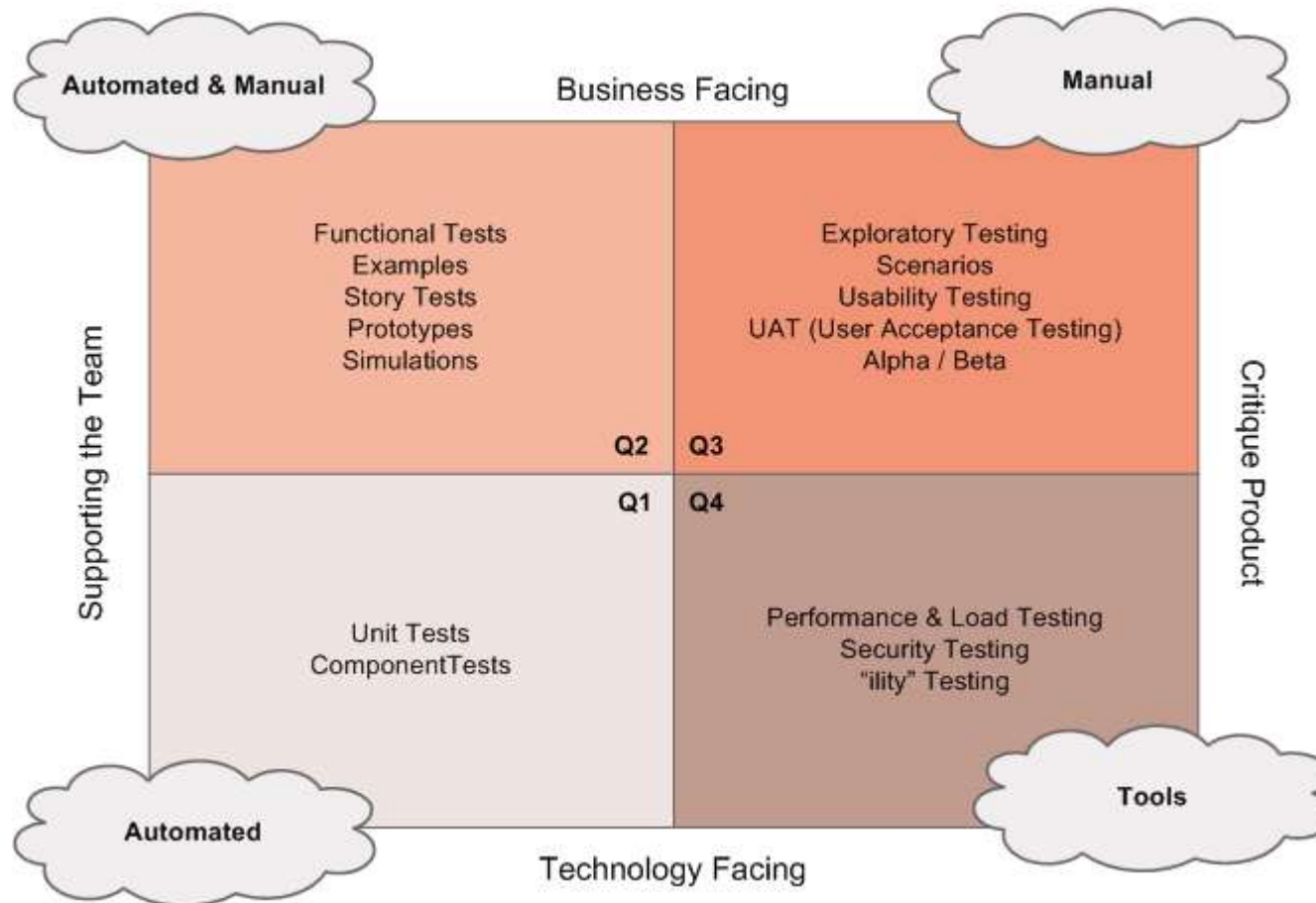
Root causes: during the project

- Risks and changes
 - Change management is missing (scope creep)
 - Impact analysis for changes is not conducted
 - Risks are not identified nor reviewed
 - Specs and agreements not updated
- Processes
 - Customer/business is forgotten and allows to be so
 - Actual work processes supported by software are forgotten to be changed

Root causes: Acceptance and finishing off

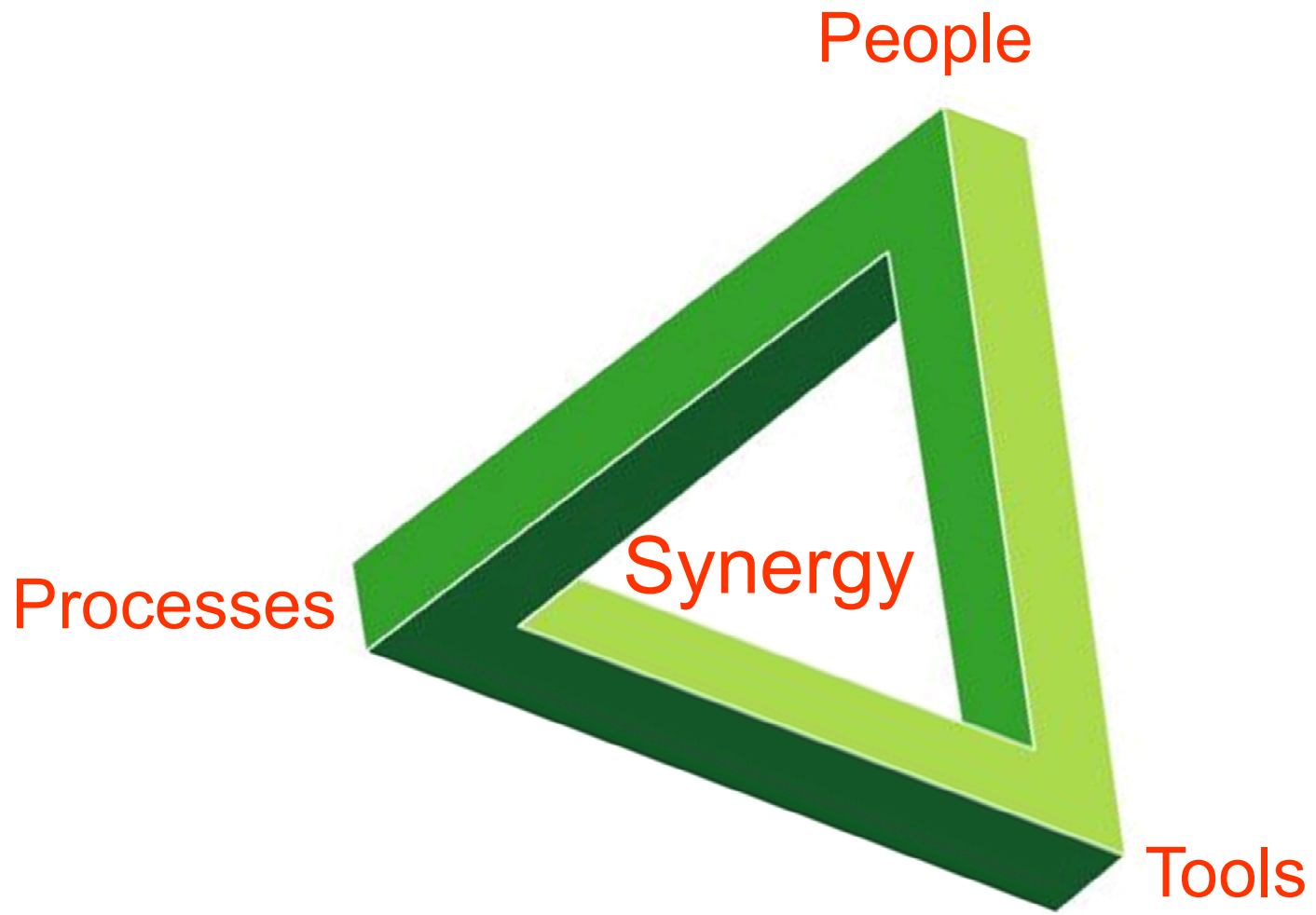
- Some requirements will never be tested
 - Usually non-functional
- Testing seen as isolated activity from development process, usually in the end of a phase
- Testing responsibilities not agreed
- Testing (especially acceptance by the customer) not planned
- Poor support for deployments and other op's
 - Op's not included in project team
 - Poor deployment and technical management guides

Agile Testing Quadrants



Conclusion

- The aim is not a risk-free project
- The aim is outcome that balances quality, functionality, budget and timeline





ASA
QUALITY SERVICES OÜ

Murphy never sleeps,
but spares those having
Plan B

Aitäh!

Maili Markvardt
maili@asaquality.ee