Industrial report:
Embedding Quality assurance in an Agile/Safe development approach

Valid 2023

Jos van Rooijen

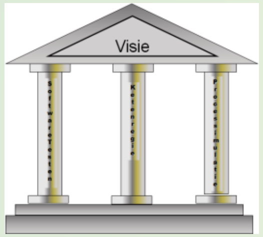
Agenda

- Introduction
- Context
- Agile / safe process
- Challenges related to quality
- Embedding the quality approach
- Future work / ideas
- Literature & Tooling

Introduction

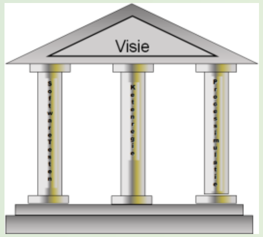
- Consultant at Huis voor Software kwaliteit
- 30 years in software testing & quality management
- Co-author several quality related books
- Test expert online magazine Computable
- Publication areas: Testing, Education and quality monitoring
- Graduation supervisor Avans university of applied science
- Member of the steering committee Valid conference
- Member of the board Dutch Testing Society

Project circumstances



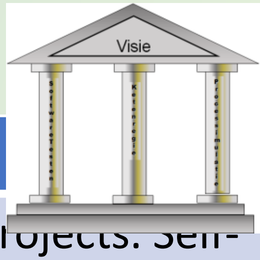
- Project with a lead time of 5 years, started in 2020
- Current development platform is end of life at 2026. Rebuilding the system from cool:gen to java platform
- No new functionality is allowed
- Social impact is huge for the public sector
- Risk of harm is tremendous
- >150 billion euro per year is going around the systems
- Key Performance Indicator(KPI) is:
  - Business continuity is priority 1
  - Business quality is also priority 1
- No. of function points >10.000
- Project budget >70 million euro’s
- Chosen development method is Agile / Safe

Context



- Agile / Safe development is common understanding inside companies nowadays
- Applied worldwide in different kind of projects
- Maturity of usage Agile varies. From 100% Agile towards Agile in name only

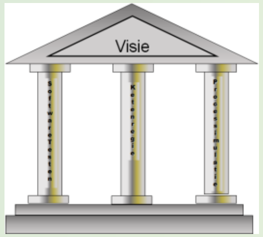
Explanation Agile terminology



Term	Description
Agile	Agile development is an iterative software-development methodology which teams use in projects. Self-organized, cross-functional teams frequently analyze circumstances and user needs to adapt projects. Scrum teams constantly improve quality in sprints with short-term deliverables
Safe	SAFe is the world's leading framework for Business Agility. SAFe integrates the power of Lean, Agile, and DevOps into a comprehensive operating system that helps enterprises thrive in the digital age by delivering innovative products and services faster, more predictably, and with higher quality
Sprint	A fixed period of time where teams are working on agreed goals. Period is 2-3 weeks
Train (ART)	The Agile Release Train (ART) is a long-lived team of Agile teams that incrementally develops, delivers, and often operates one or more solutions in a development value stream.
Agile teams	An Agile Team is a cross-functional group of typically ten or fewer individuals with all the skills necessary to define, build, test, and deliver value to their customer
ART backlog	The ART Backlog is a Kanban system that is used to capture and manage the features and enablers intended to enhance the solution and extend its architectural runway
Enablers	Enablers are backlog items that extend the architectural runway of the solution under development or improve the performance of the development value stream
Planning interval (PI)	A Planning Interval (PI) is a cadence-based timebox in which Agile Release Trains deliver continuous value to customers in alignment with PI Objectives

Source: © Scaled Agile, Inc./ www.agilealliance.org

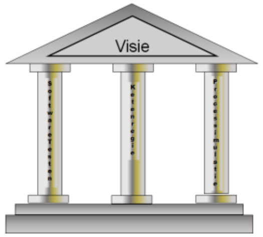
Context

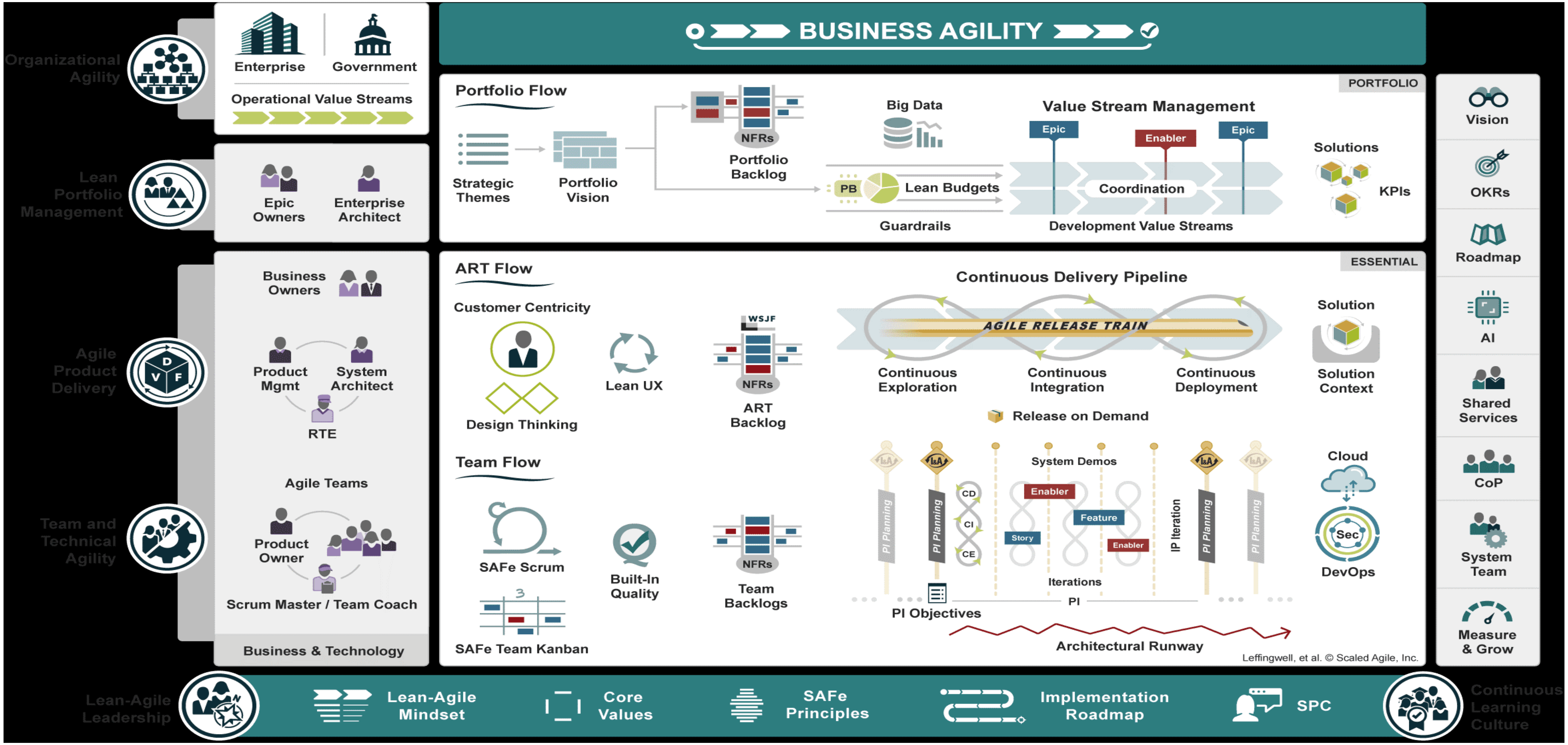


Also in this company:

- Learning applying the way of working related to Agile / Safe
- Not every department is working conform the Agile approach
- Knowledge of the Agile way of working is not really spread out in the company. Focus lies in the IT-department
- They are still learning
- Several Agile Release Trains are implemented, organized around functionality
- 8 teams are part of the this particular Agile Release Train

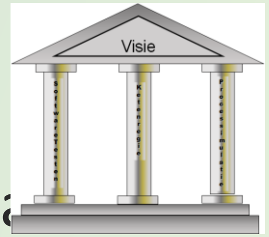
Basic flow Agile / Safe process





ScaledAgileframework.com. Safe 6.0

Challenges related to quality

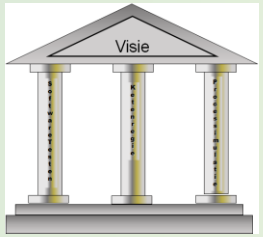


- A lot of discussion about positioning and responsibility of quality inside the tra Especially overarching matters
- Lack of knowledge
- How do you ensure that all the work is done without things falling through the cracks? It is too fragmented
- Responsibility for quality. A team, a person or the train??
- Quality activities are not aligned
- Product responsibility is not taken
- How do we keep track of the consistency and completeness of the product?
- The method is too much of a straitjacket
- Agile coaches are not aware about the impact of poor quality
- Lack of steering inside the train, despite all defined (Agile) roles
- Attitude and behavior are below par

What is the impact to the product?

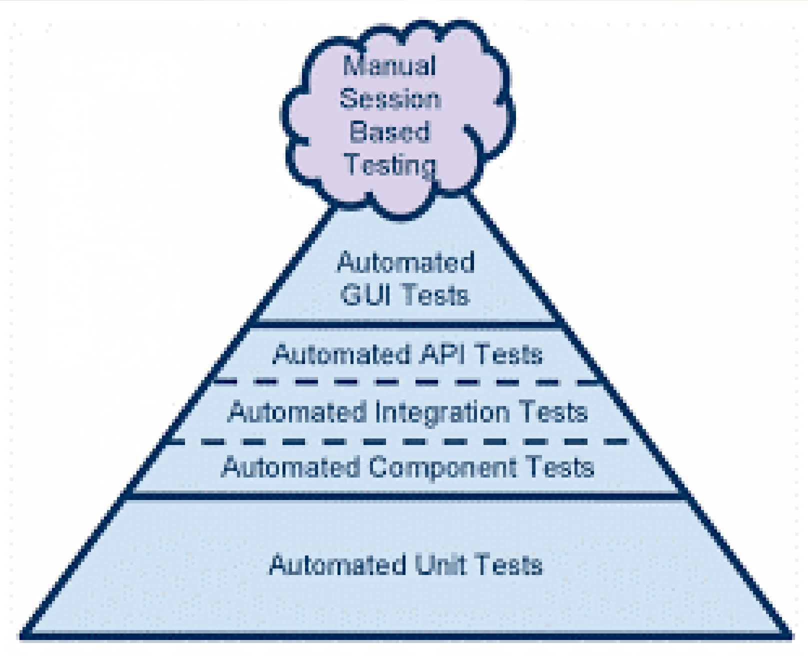
- Integrations are not always working
- Integration with technical infrastructure is not always working properly
- Integration between old and new technology is not always working quite well
- System is not working properly
- Customers faced some challenges
- Stability and continuity of the product is not guaranteed (lack of confidence)

How to deal with these issues?



Principles of the approach

- Following the testpyramid
- As early as possible
- Overarching when it is necessary
- Orchestration on overall quality
- The project is labeled as a strategic project



Manual Session Based Testing

Automated GUI Tests

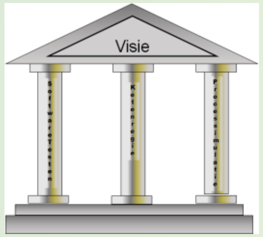
Automated API Tests

Automated Integration Tests

Automated Component Tests

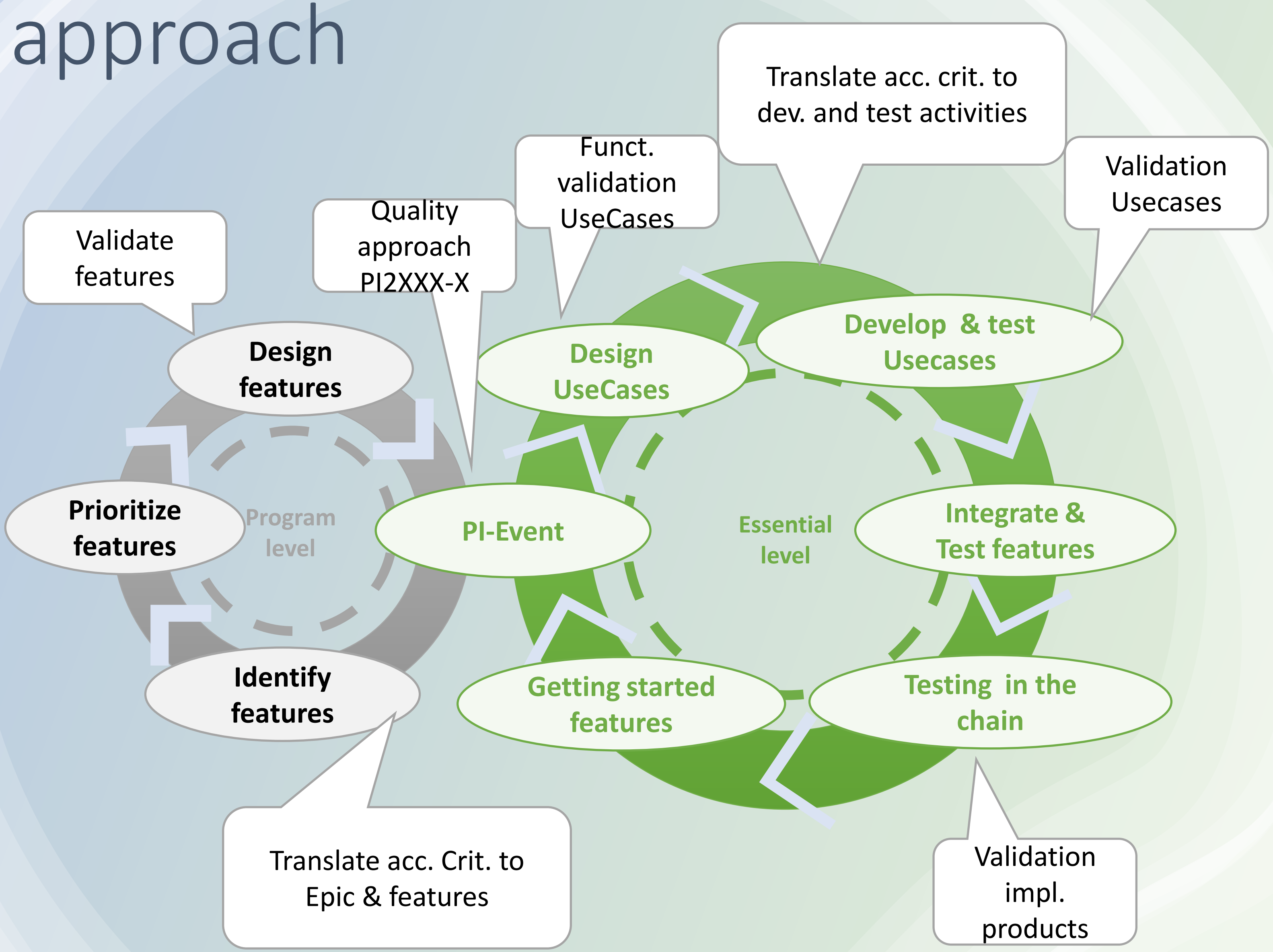
Automated Unit Tests

Our solution: a multi step approach



1. Special interventions
2. Positioning of a quality orchestrator
3. Embedding quality in each step of the train
4. Organize quality from an overall test strategy
5. Translate the overall test strategy into a test approach per feature / release together with all involved teams
6. Monitor quality progress via:
    1. Demo’s
    2. Audits
    3. Internal review
    4. Subject matter expert meeting
7. Organize a walkthrough to define completeness of work
8. Reorganize labor
9. Introduce in 2024 acceptance test-driven development

Quality approach



Validate features

Quality approach PI2XXX-X

Funct. validation UseCases

Translate acc. crit. to dev. and test activities

Validation Usecases

Design features

Prioritize features

Program level

Identify features

PI-Event

Design UseCases

Develop & test Usecases

Essential level

Integrate & Test features

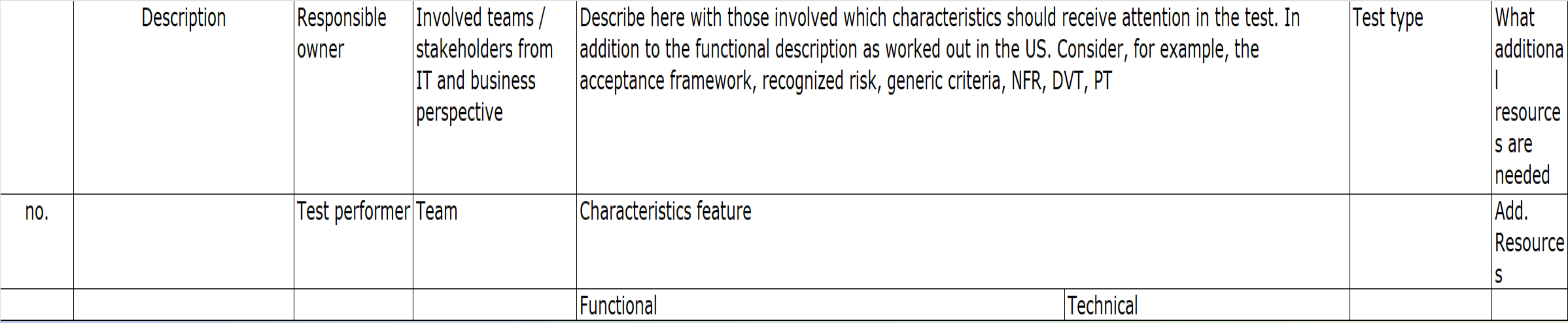
Getting started features

Testing in the chain

Translate acc. Crit. to Epic & features

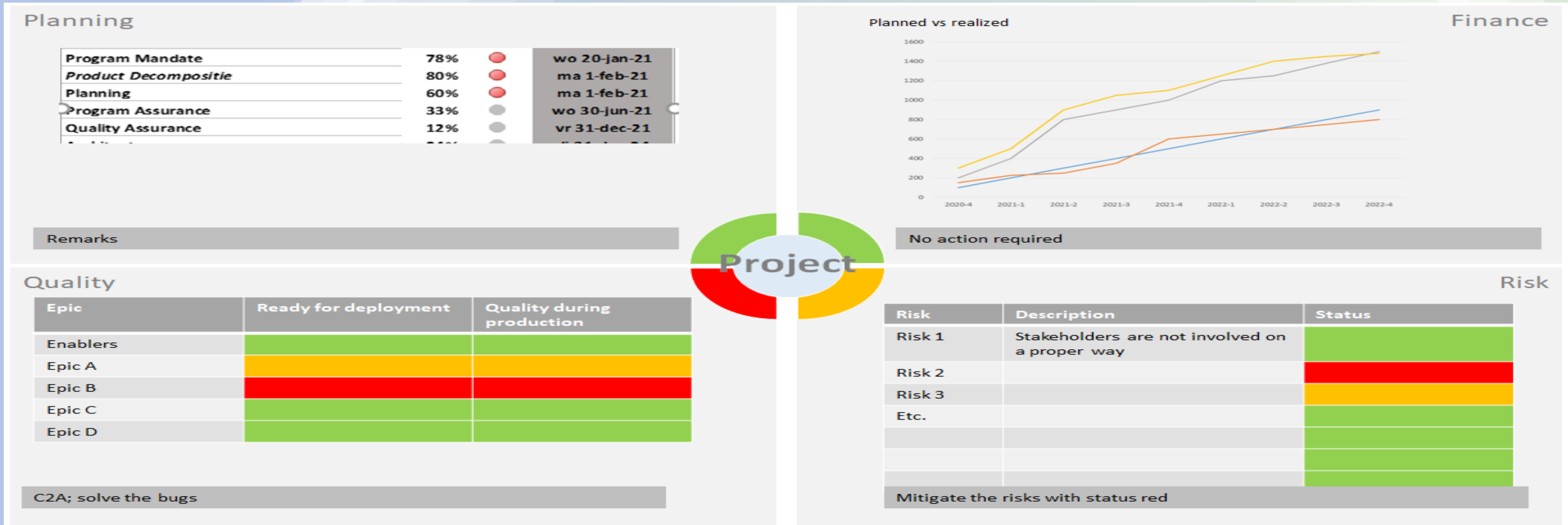
Validation impl. products

Example of a quality approach



	Description	Responsible owner	Involved teams / stakeholders from IT and business perspective	Describe here with those involved which characteristics should receive attention in the test. In addition to the functional description as worked out in the US. Consider, for example, the acceptance framework, recognized risk, generic criteria, NFR, DVT, PT		Test type	What additional resources are needed
no.		Test performer	Team	Characteristics feature			Add. Resources
				Functional	Technical		

Metrics model



Planning

Program Mandate	78%	🔴	wo 20-jan-21
Product Decompositie	80%	🔴	ma 1-feb-21
Planning	60%	🔴	ma 1-feb-21
Program Assurance	33%	⚪	wo 30-jun-21
Quality Assurance	12%	⚪	vr 31-dec-21

Remarks

Finance

Planned vs realized

No action required

Project

Quality

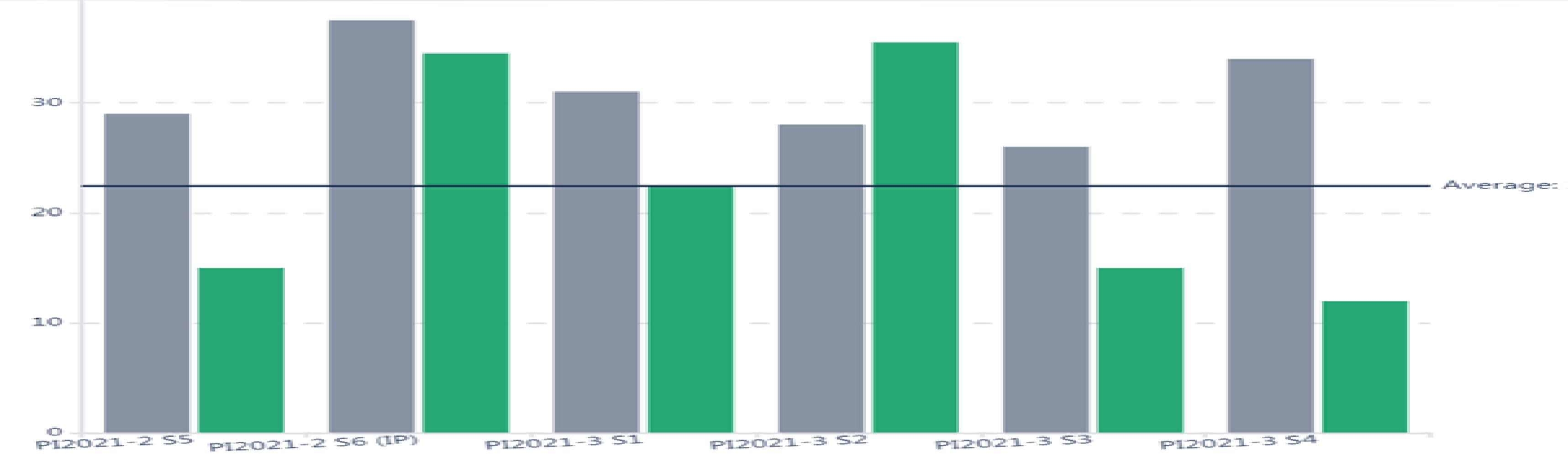
Epic	Ready for deployment	Quality during production
Enablers	🟩	🟩
Epic A	🟧	🟧
Epic B	🟥	🟥
Epic C	🟩	🟩
Epic D	🟩	🟩

C2A; solve the bugs

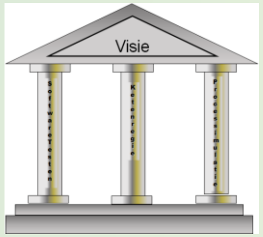
Risk

Risk	Description	Status
Risk 1	Stakeholders are not involved on a proper way	🟩
Risk 2		🟥
Risk 3		🟧
Etc.		🟩
		🟩
		🟩
		🟩

Mitigate the risks with status red



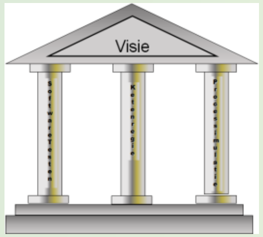
What are the results of the interventions?



We are not yet on the required level. It is a persistent situation. But, we are gaining on maturity:

- Velocity is increasing
- Planning is more realistic and more predictable
- Teams are able to reach their sprint goals. They are getting more pro-active to their environment
- Business participation is increasing and is coming more mature on quality perspective
- Right people with the right skills
- There is a more predictable flow in the train

Future work



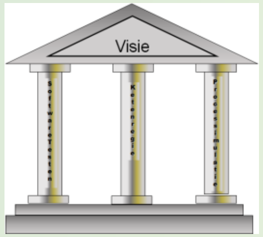
- Increasing level of test automation
- Increasing the feature and usecase phase
- Develop more smart criteria
- Increase the business participation

Literature & tooling

Literature:

1. ‘Project de Baas’- J. van Rooyen et all – 2011
2. ‘De kwaliteitsregisseur’ – Werkgroep Testregie, TestNet – 2011
3. ‘Regie van kwaliteit’- J. van Rooyen et all – Academic Service – 2012
4. ‘Niet managen maar regie voeren’- Leon Dohmen – Automatiseringsgids – 2010
5. ScaledAgileframework.com. Safe 6.0
6. ‘Agile werkt’ - H. Huijgens – Academic Service – 2012
7. ‘functional metrics’ – Nesma – 2016
8. ‘test pyramid’- MartinFowler.com’- 2018
9. www.agilealliance.org

Literature & tooling



Tooling:

1. Jira used for planning and bug handling during all steps in the Agile Release Train
2. Confluence used for document storage during all steps in the Agile Release Train
3. RobotFramework used for test automation during all steps in the Agile Release Train
4. Bitbucket is used to store and manage code, as well as track and control the changes to the code
5. Excel for generating test data
6. Jmeter to prepare and execute performance testing during acceptance test

Questions?

Thanks for your attention.

Jos van Rooijen | 0031 (0)6 -54 90 62 82 | jos@huisvoorsoftwarekwaliteit.com