Useful stepping stones in growth towards Agile testing.

Handige stepping-stones voor testers op weg naar succes in Agile context.

Kees Blokland

kees.blokland@polteq.com

https://improvement.polteq.com/ti4agile

Transition to Agile

- Many follow a stepwise approach
- Cross the river using stepping stones (*and keep going*)

Approach for transition to Agile (testing focus)

- How to outline a route towards Agile (testing)?
- How to stimulate / motivate and prevent stagnation?

- Recipe:
  - Start form an industry standard test improvement approach
  - Identify ‘Agile’ and ‘Scrum’ ingredients
  - Mix these with a lot of field experience

- Result: TI4Agile

Agile testing maturity levels

	Forming	Norming	Performing
Agile testing levels	Set the basis and the first steps towards working in an Agile manner	Adopt a process that facilitates the Agile view on working	Continuously improve the way you work by living the Agile way

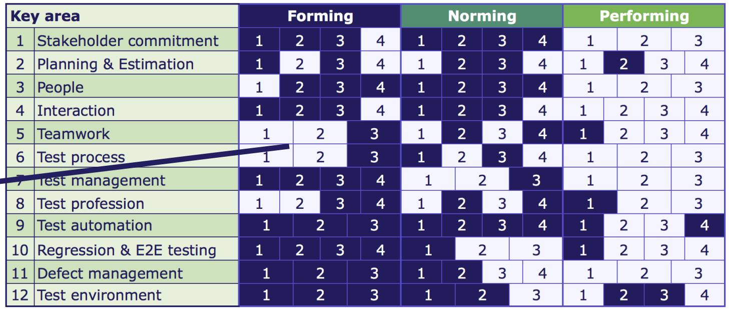
Key areas and checkpoints

Key area		Forming				Norming				Performing			
1	Stakeholder commitment	1	2	3	4	1	2	3	4	1	2	3	
2	Planning & Estimation	1	2	3	4	1	2	3	4	1	2	3	4
3	People	1	2	3	4	1	2	3	4	1	2	3	
4	Interaction	1	2	3	4	1	2	3	4	1	2	3	4
5	Teamwork	1	2	3		1	2	3	4	1	2	3	4
6	Test process	1	2	3		1	2	3	4	1	2	3	
7	Test management	1	2	3	4	1	2	3		1	2	3	
8	Test profession	1	2	3	4	1	2	3	4	1	2	3	
9	Test automation	1	2	3		1	2	3	4	1	2	3	4
10	Regression & E2E testing	1	2	3	4	1	2	3		1	2	3	4
11	Defect management	1	2	3		1	2	3	4	1	2	3	
12	Test environment	1	2	3		1	2	3		1	2	3	4

Checkpoint description on https://improvement.polteq.com/ti4agile

How to outline a route to Agile testing?

- Make a picture of the current situation using the model



Key area		Forming				Norming				Performing			
1	Stakeholder commitment	1	2	3	4	1	2	3	4	1	2	3	
2	Planning & Estimation	1	2	3	4	1	2	3	4	1	2	3	4
3	People	1	2	3	4	1	2	3	4	1	2	3	
4	Interaction	1	2	3	4	1	2	3	4	1	2	3	4
5	Teamwork	1	2	3		1	2	3	4	1	2	3	4
6	Test process	1	2	3		1	2	3	4	1	2	3	
7	Test management	1	2	3	4	1	2	3		1	2	3	
8	Test profession	1	2	3	4	1	2	3	4	1	2	3	
9	Test automation	1	2	3		1	2	3	4	1	2	3	4
10	Regression & E2E testing	1	2	3	4	1	2	3		1	2	3	4
11	Defect management	1	2	3		1	2	3	4	1	2	3	
12	Test environment	1	2	3		1	2	3		1	2	3	4

- Identify ‘gaps’

- Decide on what to do first and what to do next
  - Quick wins, what can be done now
  - Short term and mid term
  - Logical sequence as suggested by the model
  - What is on the horizon? Where to aim for?

- Implement!

How to stimulate and motivate?

- What is in it for everyone?
- 'Agile' is not the goal, it is *the benefits* of Agile
- What are benefits of important steps underway?
- When do we get such benefits?
- Mark the route with stepping stones!

Just good testing!

Key area		Forming				Norming				Performing			
1	Stakeholder commitment	1	2	3	4	1	2	3	*4*	1	2	3	
2	Planning & Estimation	1	2	3	4	1	2	3	4	1	2	3	4
3	People	*1*	2	*3*	*4*	*1*	2	3	*4*	1	2	3	
4	Interaction	*1*	2	3	4	*1*	2	3	4	1	2	3	4
5	Teamwork	1	2	3		1	2	3	4	1	2	3	4
6	Test process	1	*2*	3		1	2	3	4	1	2	3	
7	Test management	1	2	3	4	1		3		1	2	3	
8	Test profession	1	2	3	*4*	1	2	3	4	1	2	3	
9	Test automation	1	2	3		1	2	3	4	1	2	3	4
10	Regression & E2E testing	1	2	3	4	1	2	3		1	2	3	4
11	Defect management	1	2	3		1	2	3	4	1	2	3	
12	Test environment	1	2	*3*			2	3		1	2	3	4

As testers we need to make choices

What to **test** and what **not** to test

What to test **thoroughly**, what to test **superficially**

What to test **first**, what to test **later**

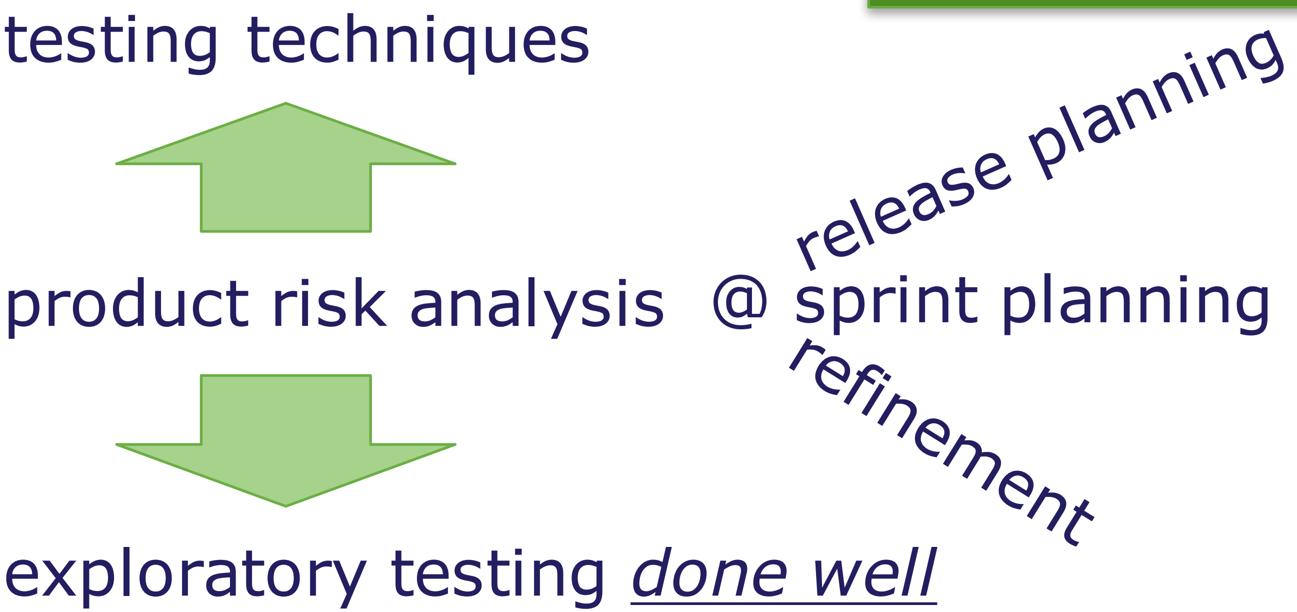
What to test **once**, what to test **each** software build

What to test by **senior** testers, what by others

What to test **scripted**, what to test **exploratory**,

… just to name a few

Just good testing!



testing techniques

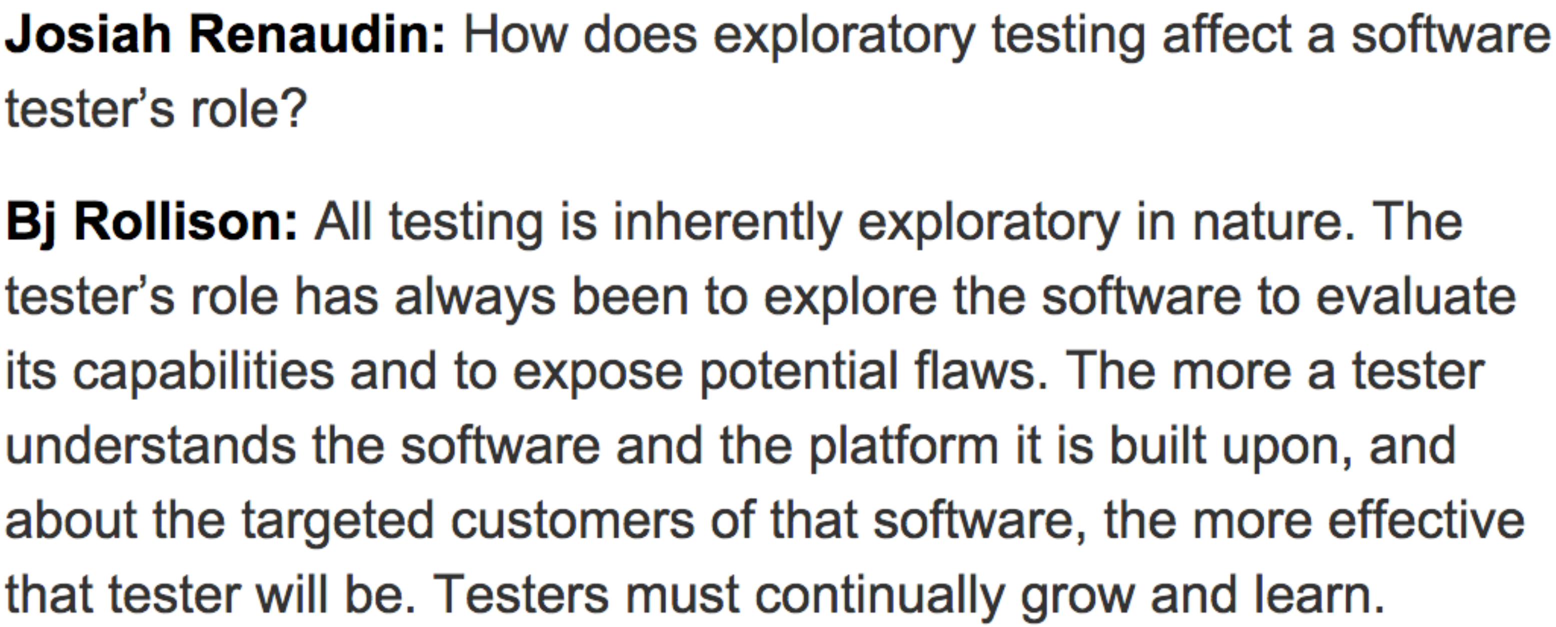
product risk analysis @ sprint planning

release planning

refinement

exploratory testing *done well*

Just good testing!



Josiah Renaudin: How does exploratory testing affect a software tester's role?

Bj Rollison: All testing is inherently exploratory in nature. The tester's role has always been to explore the software to evaluate its capabilities and to expose potential flaws. The more a tester understands the software and the platform it is built upon, and about the targeted customers of that software, the more effective that tester will be. Testers must continually grow and learn.

http://www.stickyminds.com/interview/things-really-matter-testing-today-and-tomorrow-interview-bj-rollison

Just good testing!

Benefits

Getting the best testing in the time available

- Most serious bugs found
- Fit for purpose

Leadership

- Wait-and-see attitude
- Need to pull one's socks up when sprint goals are at risk
- (Lack of) soft skill development
- “Who takes the testing role for this story?”
- Specialisms limit multi-employability

Leadership

http://www.stickyminds.com/article/seven-signs-great-agile-leadership

"Agile teams are self-organizing, which means they do not need supervisors or any explicit leader—at least in theory. But they do need leaders to create a shared vision of what the product will be. Without that, you will get an inconsistent product, which means low quality. In other words: If you are a quality professional, you need to care about leadership.

Having an agile team means that anyone can step up … including you. An agile leader could be a team member who is getting stories ready to review in the backlog, or the person who is giving a sprint demo and may be showing the work of multiple team members. Anyone on the team may lead for a task, and the very next day, sprint, or month, slip out of leadership and go back to everyday job duties. Team members can both lead and follow. However, the organic agile leader may need to work harder initially to collaborate with team members who are not used to transitioning between leading and following."

Tester role

From quality assurance to quality assistance

Leadership

Benefits

Getting the right product developed
Team members master multiple disciplines

Cross functional team in one location

Practice example

		Forming				Norming				Performing			
1	Stakeholder commitment	1	2	3	🟩	1	2	3	4	1	2	3	
2	Planning & Estimation	1	🟩	3	4	🟩	2	3	4	1	2	3	4
3	People	1	2	3	4	1	2	3	4	1	2	3	
4	Interaction	1	2	3	🟩	1	2	3	4	1	2	3	4
5	Teamwork	🟩	🟩	3		🟩	2	🟩	4	1	2	3	4
6	Test process	1	2	3		1	🟩	3	🟩	1	2	3	
7	Test management	1	2	3	4	1	2	3		1	2	3	
8	Test profession	1	2	3	4	1	2	🟩	4	1	2	3	
9	Test automation	1	2	3		1	2	3	4	1	2	3	4
10	Regression & E2E testing	1	2	3	4	1	2	3		1	2	3	4
11	Defect management	1	2	3		1	2	🟩	4	1	2	3	
12	Test environment	1	2	3		1	2	3		1	2	3	4

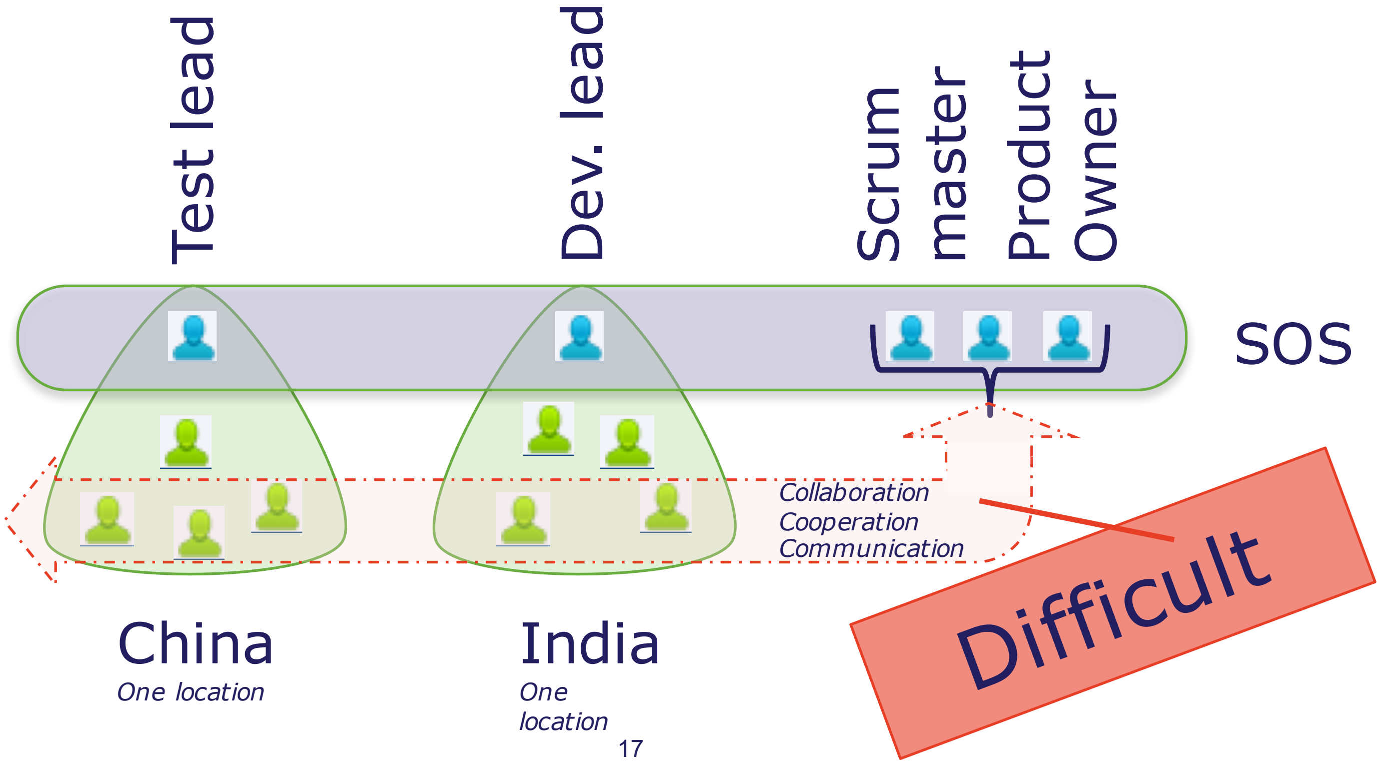
Cross functional team in one location

Agile principles

1. Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software
2. Welcome changing requirements, even late in the development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with preference to the short time scale.
4. Business people and developers must work together to the short time scale.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. **Simplicity - the art of maximizing the work not done - is essential.**
11. **The best architecture, requirements and designs emerge from self-organising teams.**
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts behaviour accordingly.

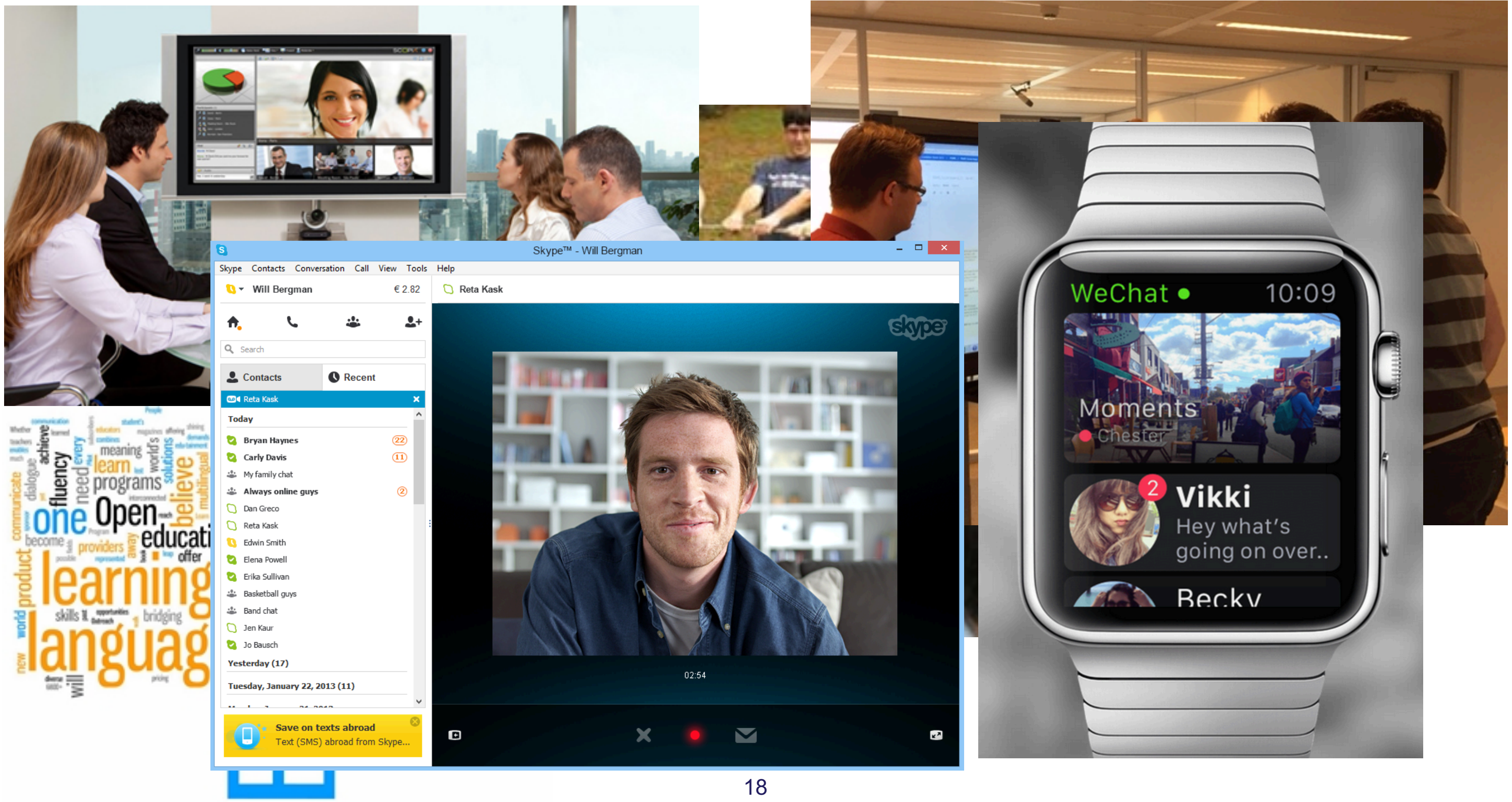
Cross functional team in one location

Practice example



© 2016

Cross functional team in one location



Cross functional team in one location

Benefits

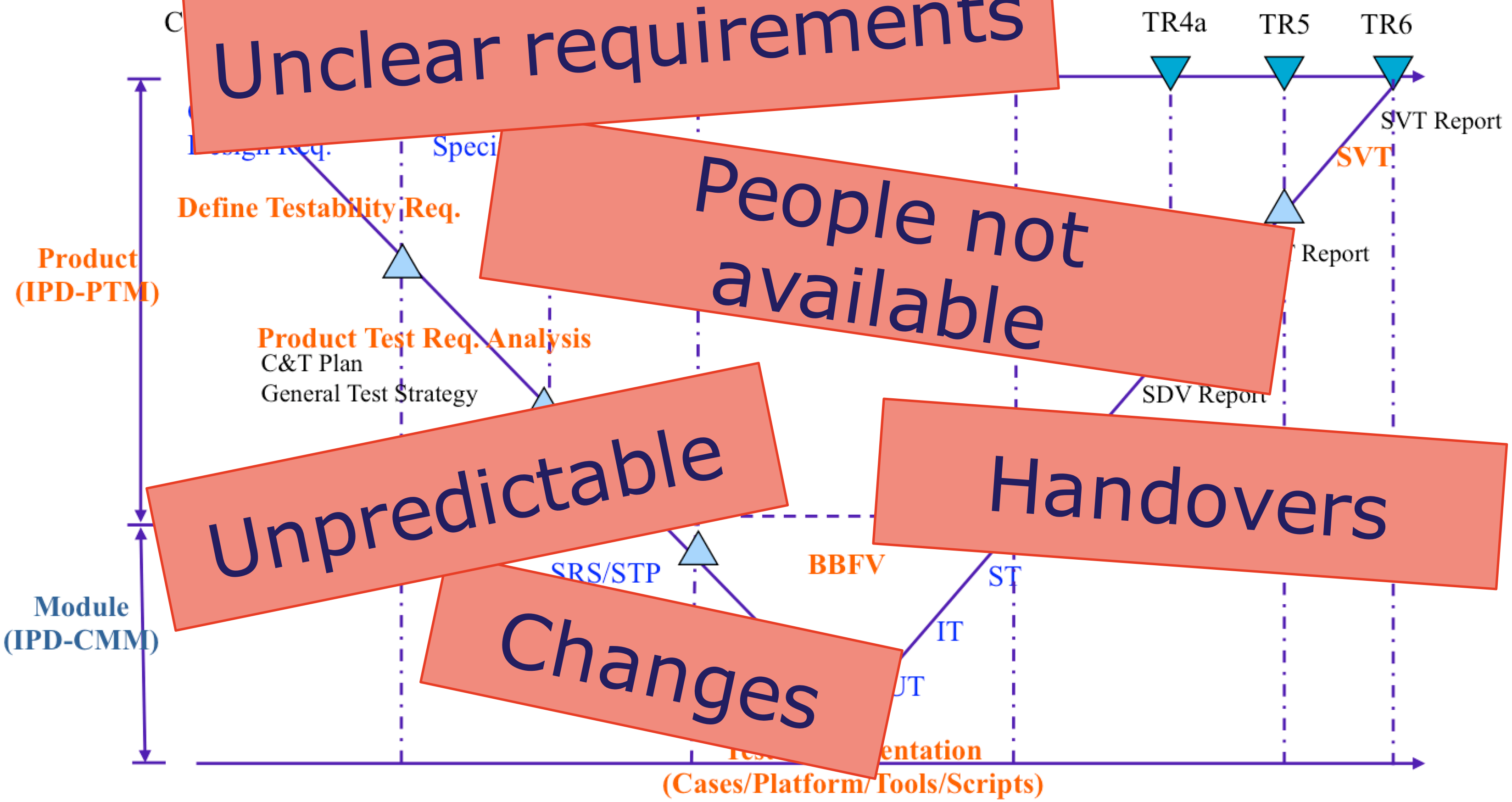
Enables Agile

- Teamwork
- Interaction
- Lean documentation

Increases team velocity

Simplified process – 1

Practice example



Unclear requirements

People not available

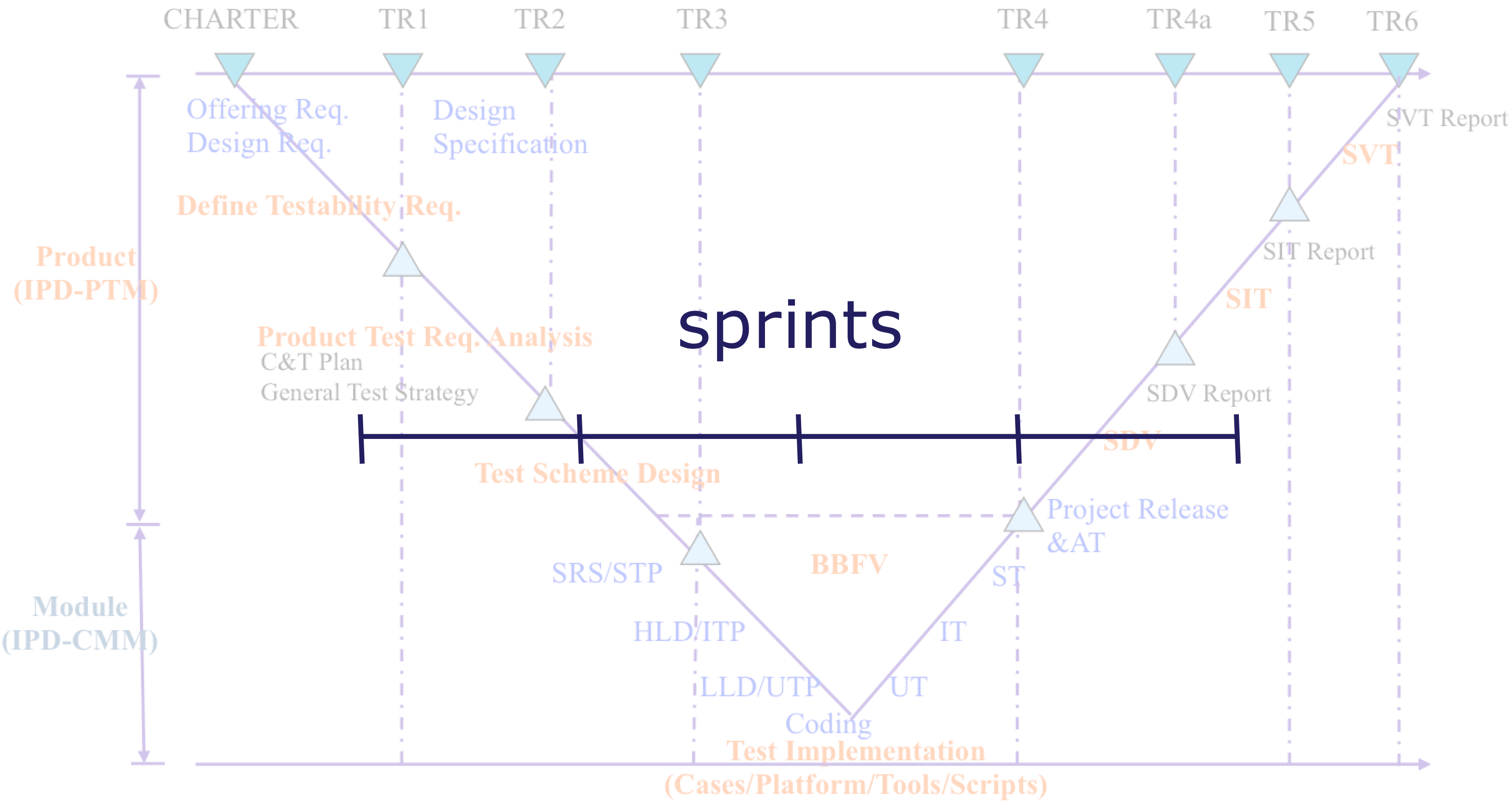
Unpredictable

Handovers

Changes

© 2016

Simplified process - 1



CHARTER TR1 TR2 TR3 TR4 TR4a TR5 TR6

Offering Req.
Design Req.

Design
Specification

Define Testability Req.

Product
(IPD-PTM)

Product Test Req. Analysis
C&T Plan
General Test Strategy

sprints

Test Scheme Design

SRS/STP

HLD/ITP

LLD/UTP

Coding

Test Implementation
(Cases/Platform/Tools/Scripts)

Module
(IPD-CMM)

BBFV

UT

IT

ST

Project Release
&AT

SDV

SDV Report

SIT

SIT Report

SVT

SVT Report

© 2016

Simplified process - 1

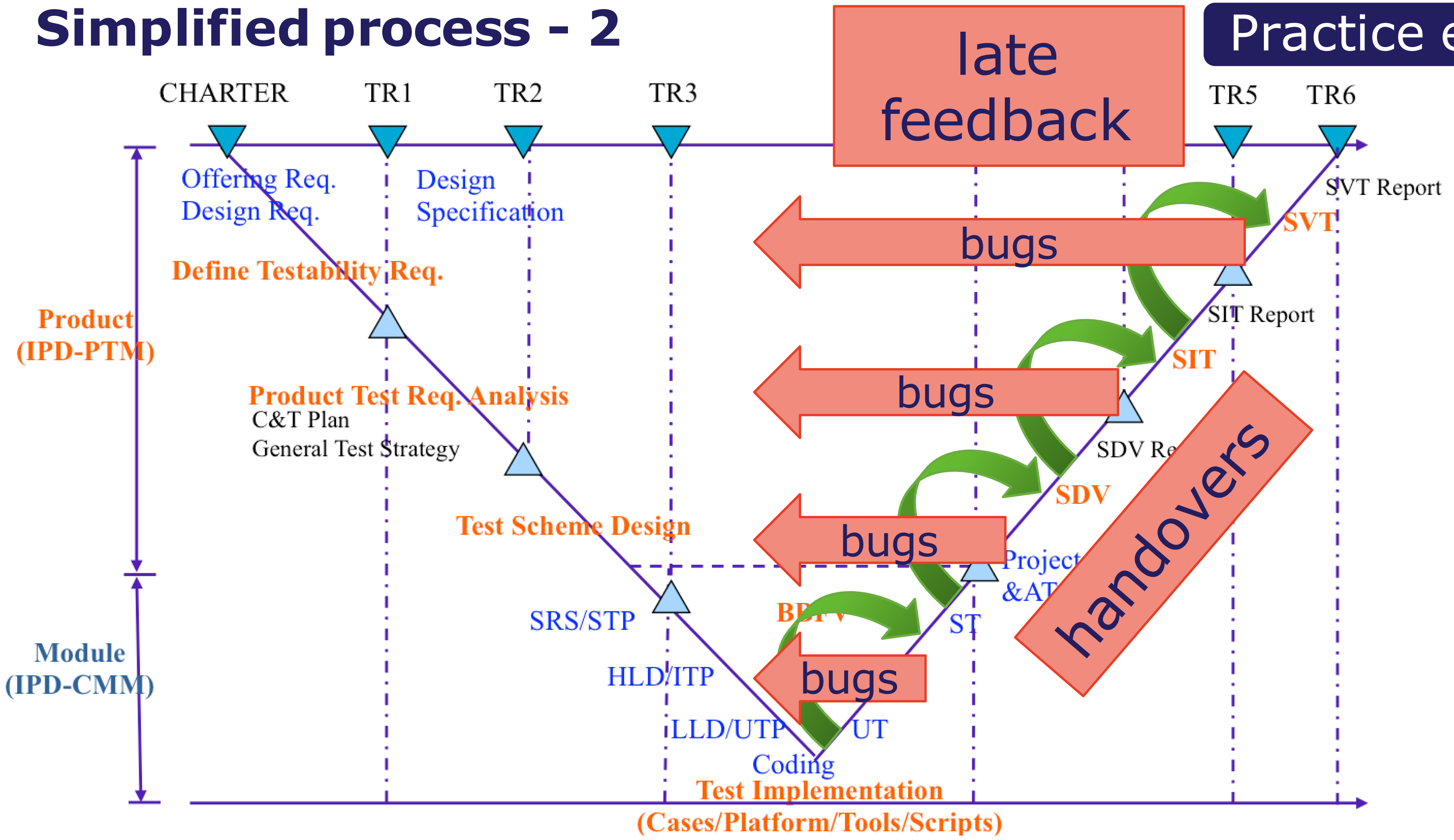
Benefits

Increases predictability of project results

- Better scope control
- Less delays
- Easier to deal with changes

Simplified process - 2

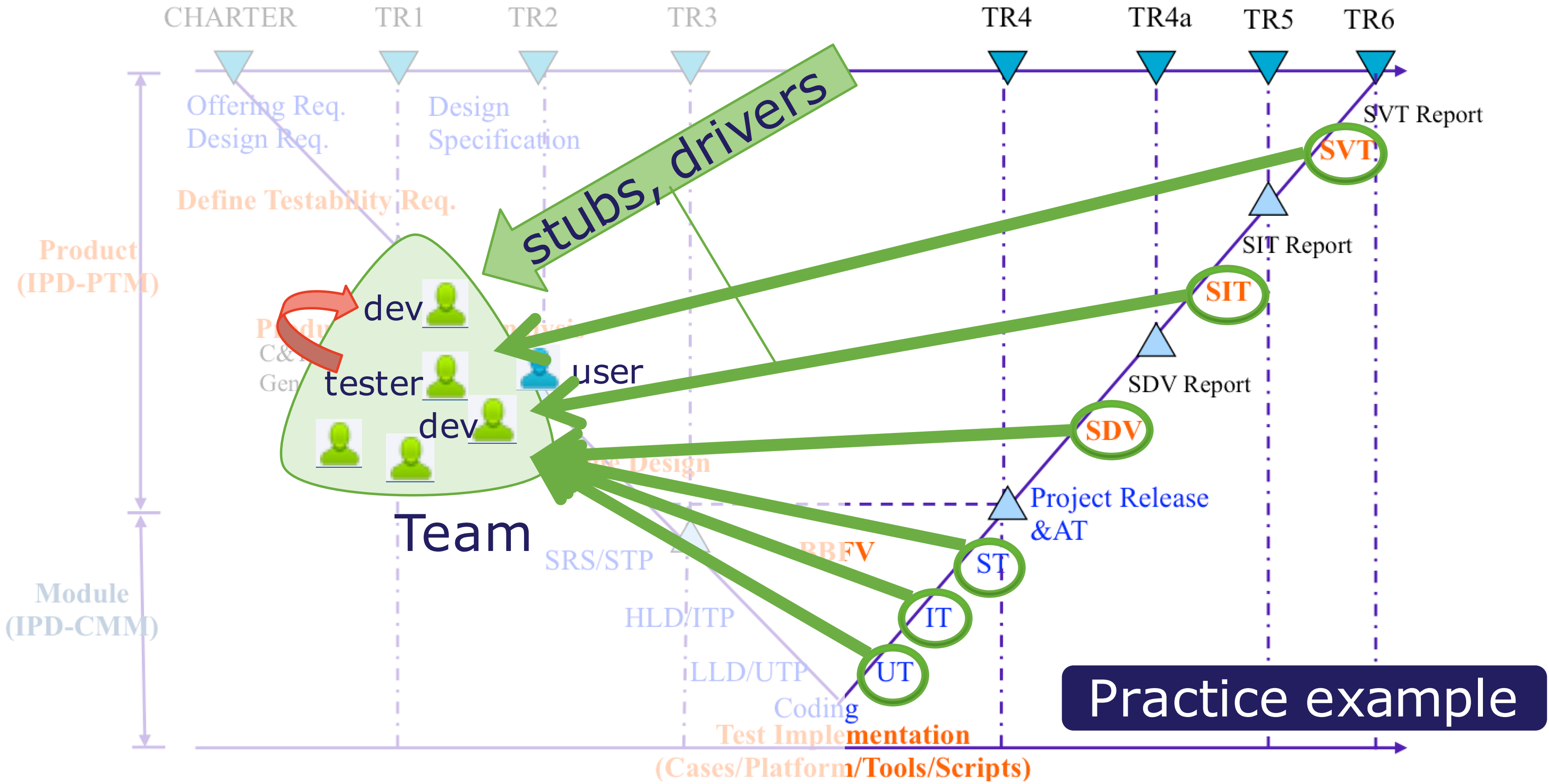
Practice example



© 2016

polteq

Simplified process - 2



Practice example

Simplified process - 2

Benefits

Reduces 'waterfall-ish' behaviour
Enables lean documentation
Increase project predictability

Agile (?) stakeholders

Practice example

More features are pushed in the schedule than seems feasible.
Priorities are there, but still **everything MUST be delivered**
→ This prevents team to feel committed to a sprint (no believe it can be done)

Limit each sprint backlog to amount that the team can deliver 'potentially shippable' by the end of the sprint

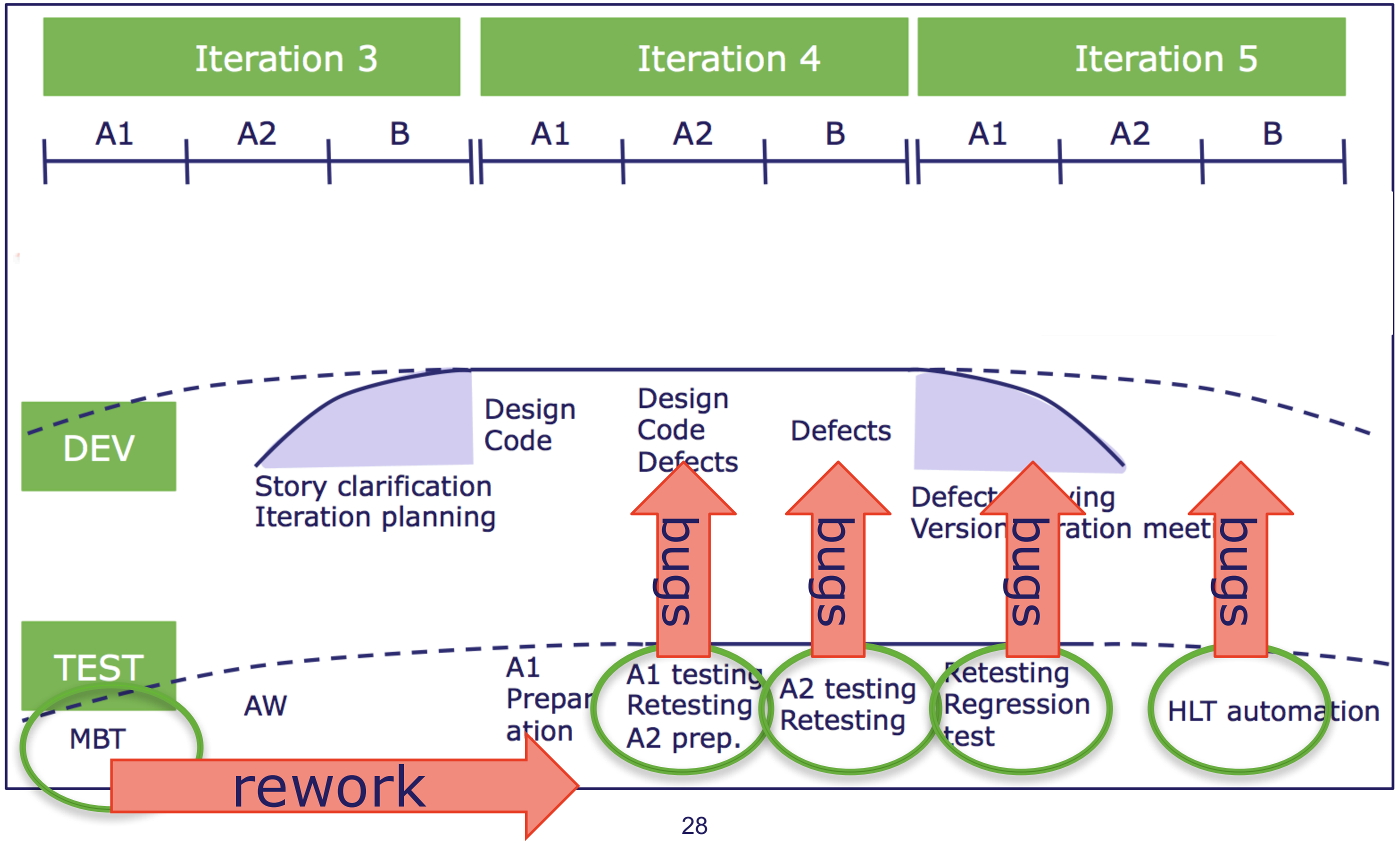
Agile (!) stakeholders

Benefits

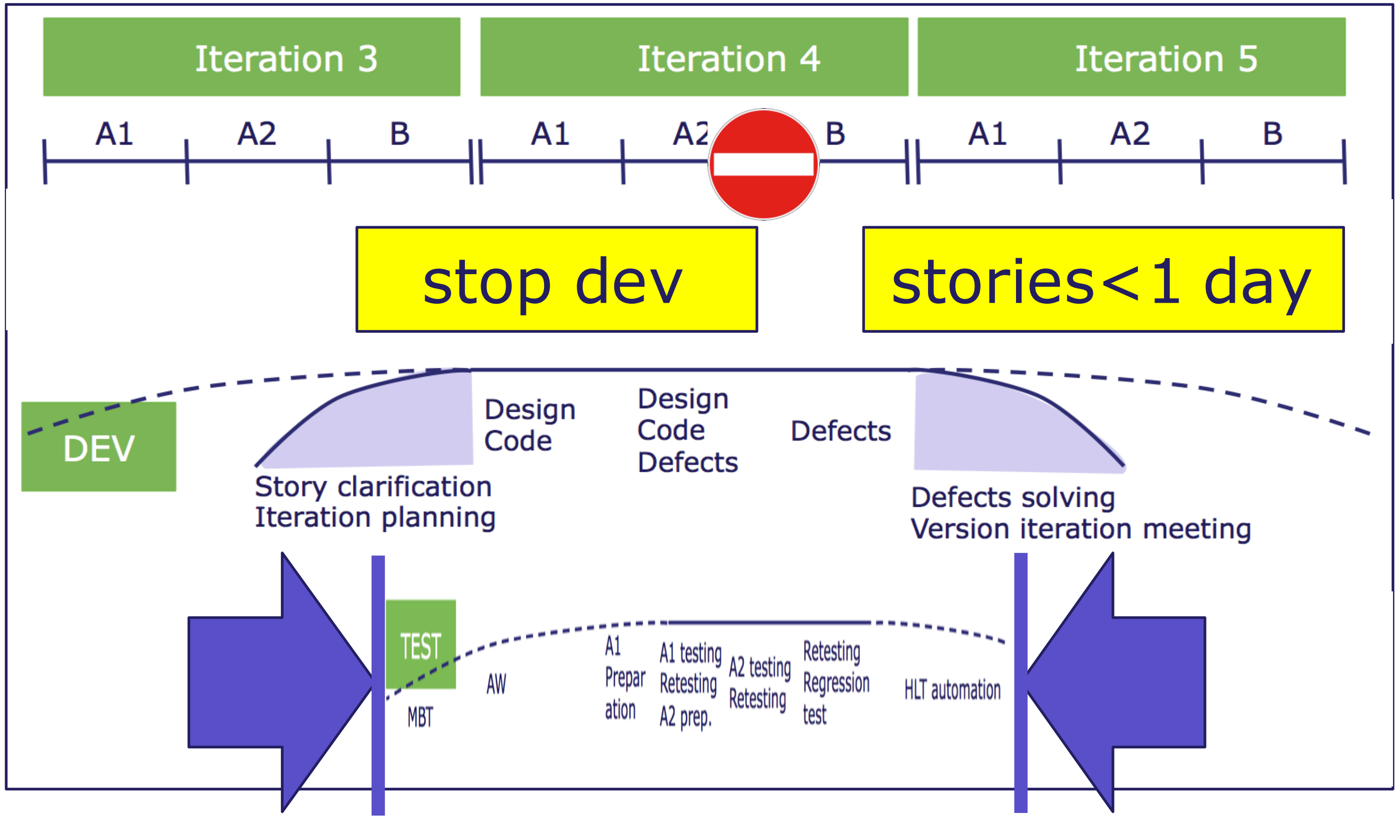
Increase of the predictability of the release
Increase of team efficiency
Increase of amount of business value per release

Finish testing within the iteration

Practice example

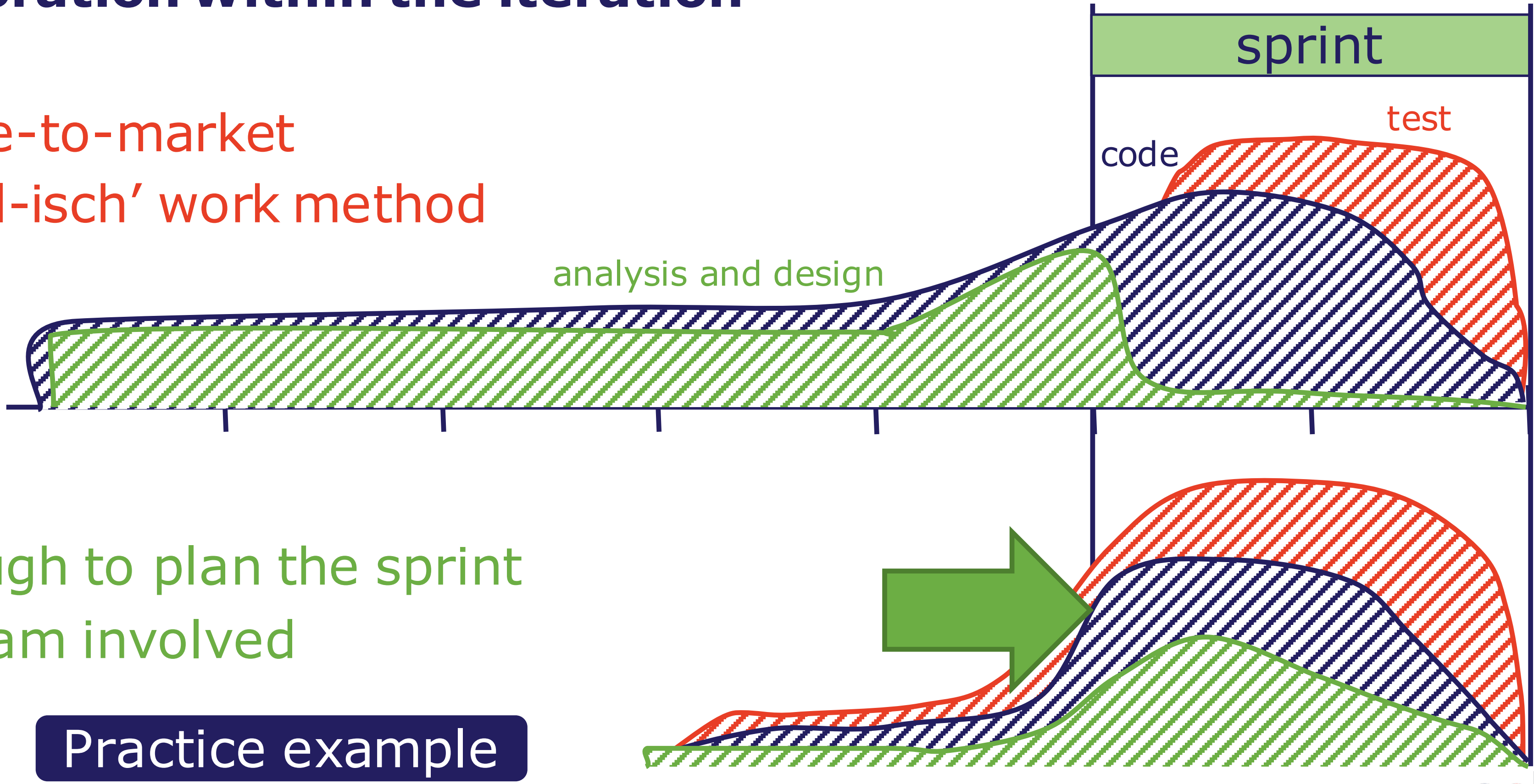


Finish testing within the iteration



Story elaboration within the iteration

- Long time-to-market
- 'Waterfall-isch' work method



- Just enough to plan the sprint
- Whole team involved

Practice example

Concentrate work within the iteration

Benefits

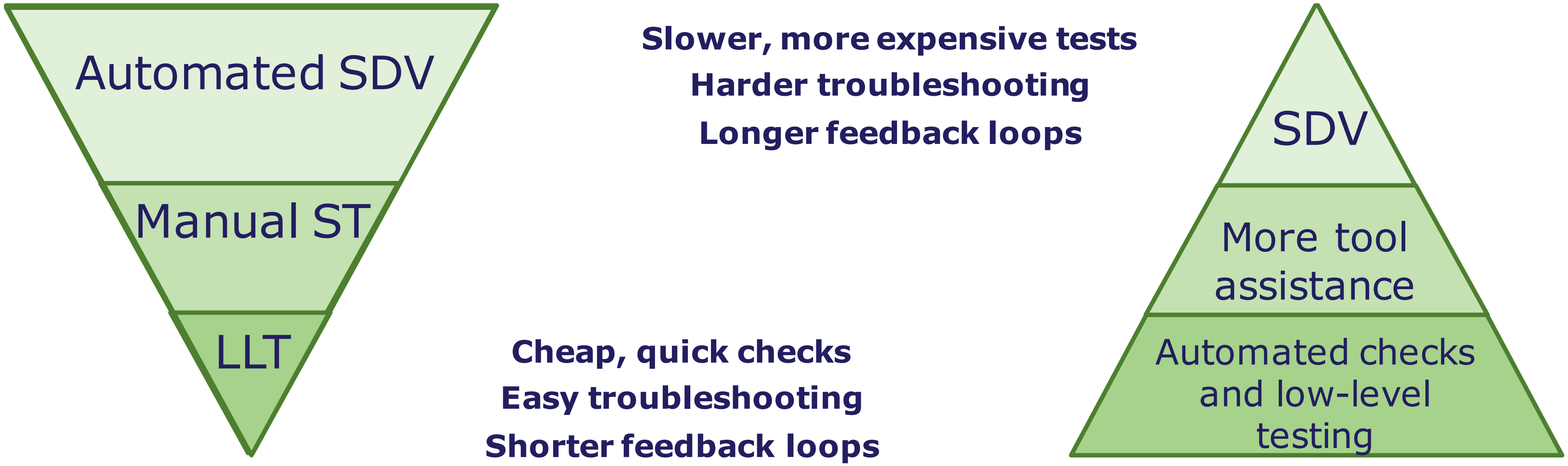
Creates flow in the team

Faster reaction on market

Potentially shippable product at the end of the sprint

Automated testing and checking

Practice example



Where coverage is

Automated SDV

Manual ST

LLT

Slower, more expensive tests
Harder troubleshooting
Longer feedback loops

Cheap, quick checks
Easy troubleshooting
Shorter feedback loops

SDV

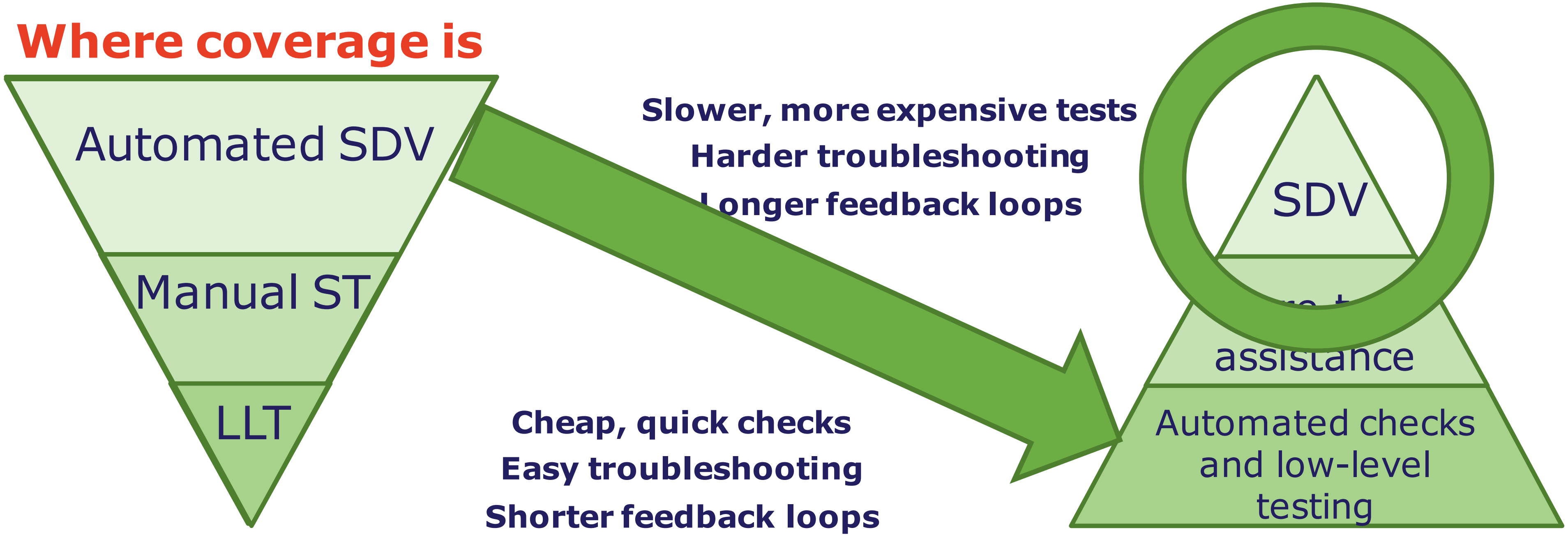
More tool assistance

Automated checks and low-level testing

Where coverage should be

Automated testing and checking

✔ Develop ‘exploratory’ style SDV using tools where appropriate



Where coverage is

Slower, more expensive tests

Harder troubleshooting

Longer feedback loops

Cheap, quick checks

Easy troubleshooting

Shorter feedback loops

uld be

✔ Move responsibility of automated checks to development teams

Automated testing and checking

Benefits

Increases team velocity
Enables sustainable team velocity
Keeps control on quality

Agile (testing) transition

Conclusion

- Stepping stones help crossing the river
- Use them and keep on moving
- Like antibiotic therapy: you need to complete it

https://groengeloven.files.wordpress.com/2014/04/ijsbeer-springt.jpg

Agile (testing) transition

Conclusion

- Stepping stones help crossing the river
- Use them and keep on moving
- Like antibiotic therapy: you need to complete it

- Recognize stepping stone benefits and motivate people based on it

- Experience shows: it's worth it!

The best testing in the time available
Increased predictability
Increased productivity

Useful stepping stones in growth towards Agile testing.

Handige stepping-stones voor testers op weg naar succes in Agile context.

Kees Blokland

kees.blokland@polteq.com

https://improvement.polteq.com/ti4agile