Plan-Driven vs. Agile

Maaret Pyhäjärvi

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Outline

- Plan-Driven vs. Agile in Development and Testing
- Experiences in Exploratory Testing
- Comparing Plan-Driven and Agile Testing Ideals

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile in Development and Testing

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven and Agility - Values

Source: Agile Alliance website <http://www.agilealliance.com>

Plan-driven Values
Processes and tools
Comprehensive documentation
Contract negotiation
Following a plan

Agile Values
Individuals and interactions
Working software
Customer collaboration
Responding to change

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile Methods

Plan-driven

- Well-understood, repeatable processes
- Plan as the center of the project
- Plan is made to achieve goals (requirements)
- Follow-up on:
  - Progress
  - Deviations from planned
- Corrective actions brings one back to the planned state

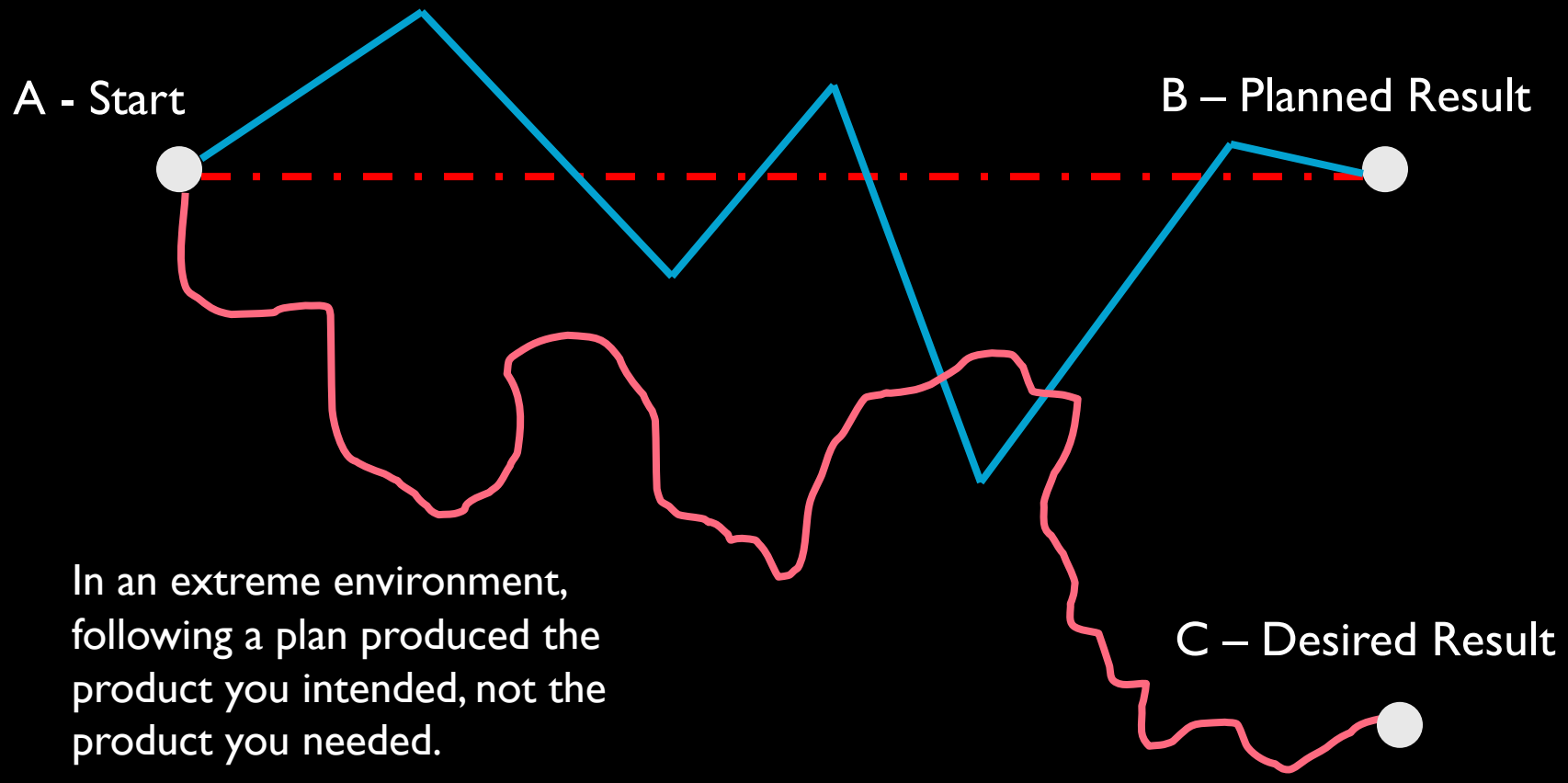
Agile

- Complex and unanticipated processes
- Results as the center of the project
- Changes can't and shouldn't be avoided.
- Nature of software development includes continuous learning.
  - Learning changes the plan

Often compared *heavy* vs. *light* and *formal* vs. *informal* – these miss the main difference!

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile - Rationale

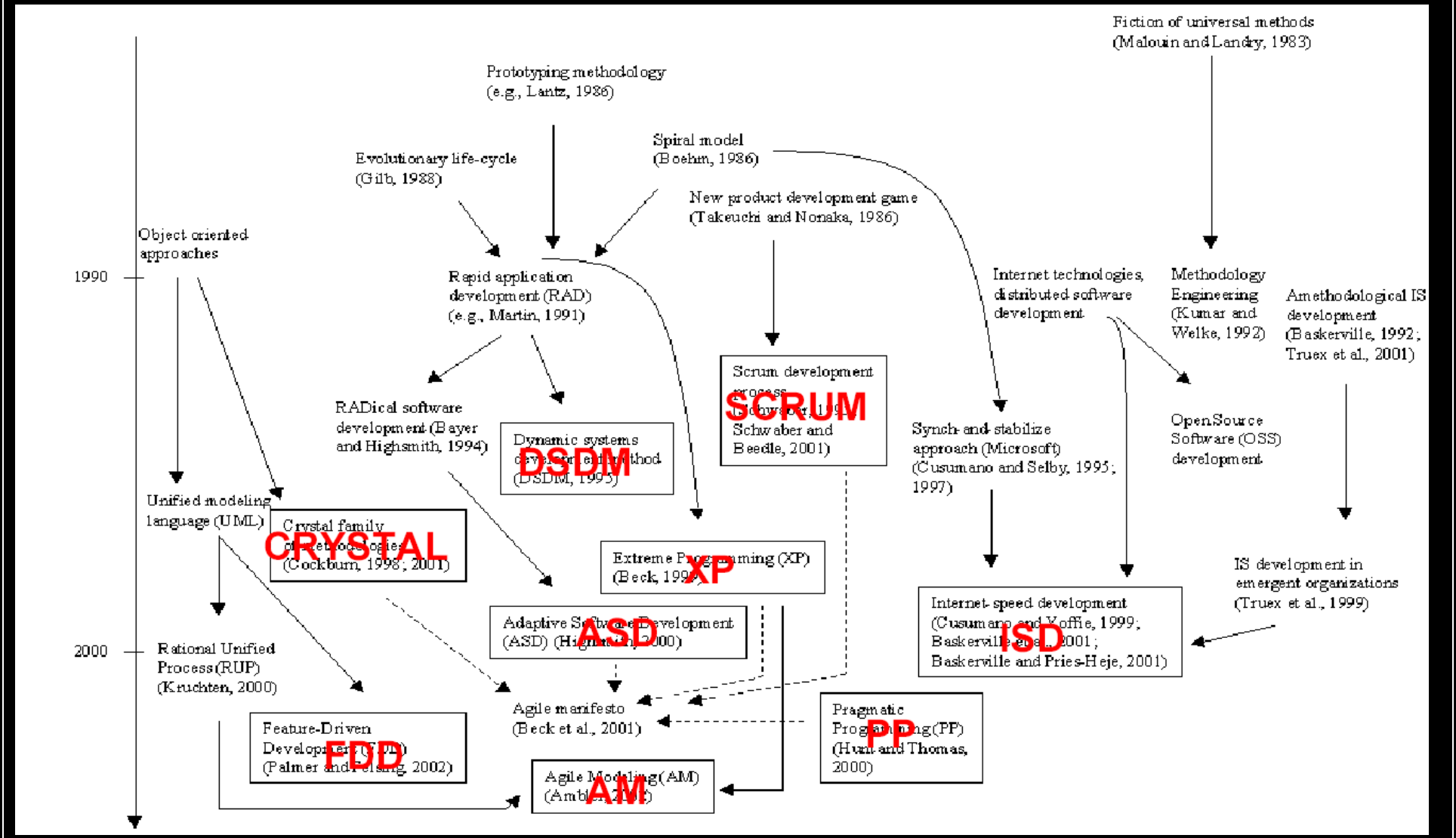


In an extreme environment, following a plan produced the product you intended, not the product you needed.

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

The World of Agile Methods

Source: Abrahamsson, P. et al. 2003. New Directions on Agile Methods: A Comparative Analysis. In *ICSE 2003*.



retp@mas.to

Agile Testing

- Two main breeds:
  - Exploratory Testing
    - Originates from testing community
    - Addresses ”system level testing”
  - Extreme Testing
    - Originates from development community (Extreme Programming)
    - Addresses ”unit level testing”
    - Requires higher-level tests (”acceptance tests”) but leaves out details of how these are included in process

Exploratory Testing vs. Ad Hoc Testing

- **Ad hoc testing**: Test execution in which fault finding has not been prepared for: testing is not planned, its goals are not identified, there are are no expectations for results and randomness guides the activity.
- Some people divide ad hoc testing in directed and undirected ad-hoc testing, in which directed ad hoc testing is used as a synonym for exploratory testing.

Mindset to Testing Impacts Test Planning and Control

- Approaching the problem from two perspectives:
  - Identifying the differences in practices used in projects
    - Based on hands-on experiences
  - Comparing two exemplary cases - Ideals
    - Plan-driven testing approach as described in ISEB Foundation & Practitioner syllabi
    - Exploratory testing approach as described in articles and training materials by James Bach & Cem Kaner

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Experiences in Exploratory Testing

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Three Depths of Exploratory Testing

ET Session
- ET as a technique

ET Team
- ET as a group's practice

ET Project
- ET as testing process

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Experiences from ET Sessions

- Exploratory Testing Sessions used during a one-year period
  - From once-a-week to twice-a-week
  - Session length 1,5 hrs (tried 1-3 hrs) and 10 people involved at a time
- A planned approach
  - Not within the master test planning but short-term – a week at a time
  - Goal or theme for each session
  - Metrics gathered

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Challenges in Getting Started

- Proving the usefulness
  - Ability to find new issues planned tests would not find
  - Good use of resources – measured effort/bug and compared ET (Exploratory testing) and ST (Planned system testing)
- Finding the right people
  - Characterictics guided the selection:
    - Naturally curious
    - Not too pattern-oriented but willing to think out-of-the-box
    - Willingness – not for everyone's taste

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

What Happened in ET Sessions

- In session
  - Briefing
    - Goal/Theme
    - Approach
  - Pair testing
  - Testing and taking notes
- Outside session
  - Reporting found bugs
  - Planning for session contents
- Challenges:
  - Coverage measurement
  - Tooling

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

A Positive Experience - Reasons

- Metrics showed the approach both effective and efficient
- Ability to acceptance test 3rd party software that could not be planned in detail
- Ability to assess how product stability (non-repro bugs) progresses
- Ability try out more combinations of external equipment easier as the loaning was short-term
- Positive impact on social relationships between testers in different projects
- Skills and tricks of the trade exchanged
  - Test data and ideas from other testers

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

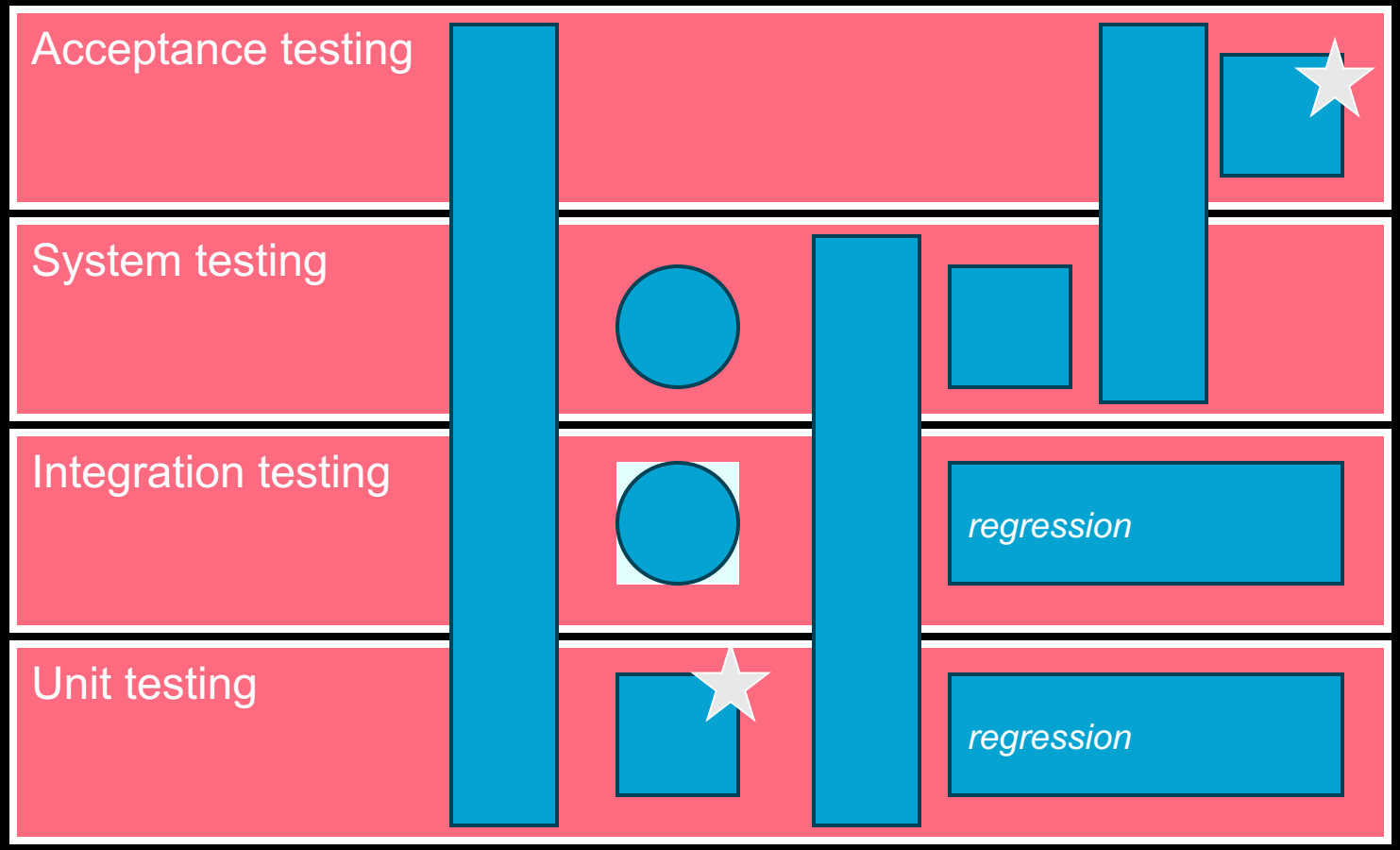
Experiences from an ET Project

- Experiences from one exploratory testing project

- Setting up expectations for test levels and pacing
- Test strategy
- Tasks, resources and schedules
- Communication with development
- Reporting and metrics

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Test Levels, Phases and Types



https://www.linkedin.com/in/maaret/ @maaretp@mas.to

In-Synch vs. Off-Synch Testing

- Length of preparation needed sets the pace of activities
  - *In-synch* testing comprises the testing tasks that can be done in the same pace as the development proceeds, in continuous fashion
  - *Off-synch* testing refers to the typical view of planned and prepared testing that takes too much time to be integrated into the daily/weekly pace of the development
- Addressing the pace
  - brings structure to agility reminding different timescales of getting things done in testing
  - brings flexibility to plan-driven addressing tasks that shouldn't be planned separately

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Test Strategy

Plan-Driven

- High-level goals set and documented before details
- Address changes and customizations to strategy in the test plan
- Separation of product/service view and project/maintenance view
- Early test design

Agile

- High-level goals set and communicated before details
- Strategy changes to accommodate the needs
- Not planning ahead requires keeping the product view in mind all the time
- On-time test design

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Tasks, Resources, Schedules

Plan-Driven

- Detail in tasks and their estimates
- Added detail in more specific plans, exceptions to plan noted
- Tends to have more variance in the expected skill set
- Separation (independence) makes most tasks off-synch

Agile

- Fixed number of resources
- Test case architecture as a general plan and reporting structure
  - Connect to dashboard, qualitative assessment
- Collaboration capability essential
- Structure to difference for in-synch/off-synch

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Communication with Development

Plan-Driven

- Independent testing
- Focus on planned features, checking realization on a test release document
- Requirements specifications

Agile

- Collaborative testing
- Focus on what is under construction, tips and hints on a test release document
- Implementation tasks for the near future

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Reporting and Metrics

Plan-Driven

- Test case – based
  - Aiming for quantification
  - Requirement traceability
- Planned vs. realized effort

Agile

- Test case architecture – based
  - Qualitative assessment on dashboard
- Use of hours as in-project feedback mechanisms for quality of testing
  - Improvement of testing skills and focus of strategy

Bugs in an essential role in both

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Comparing Plan-Driven and Agile Testing Ideals

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile in Testing (1/6)

Source: Pyhäjärvi, M. & Rautiainen, K. 2004. Integrating Testing and Implementation into Development. Engineering Management Journal, June '04

Aspect	Plan-driven testing	Exploratory testing
Order of test execution	Low-level tests are executed before high-level tests.	Testing should take place on all levels in an order that makes most sense.
Role of test documentation	If documentation does not exist, testing is not done - aim at comprehensive unambiguous documentation.	Testing is documented to the extent needed weighing together costs and benefits.
What constitutes start of testing	Starting test planning with late execution is testing.	Test planning is not start of testing, if execution can't be combined in an exploratory fashion.

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile in Testing (2/6)

Source: Pyhäjärvi, M. & Rautiainen, K. 2004. Integrating Testing and Implementation into Development. Engineering Management Journal, June '04

Aspect	Plan-driven testing	Exploratory testing
Focus in controlling the testing	V-model and size of testable items on each level.	Depth of testing for the system currently at hand based on its maturity.
View of test strategy	Seen as something above test project plans, combining several projects.	Seen as the projects tangible testing approach combining test techniques to quality criteria, project environment and product elements.
Test design	Early test design emphasized.	On-time test design emphasized.

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile in Testing (3/6)

Source: Pyhäjärvi, M. & Rautiainen, K. 2004. Integrating Testing and Implementation into Development. Engineering Management Journal, June '04

Aspect	Plan-driven testing	Exploratory testing
Test cases	Test case can be executed and interpreted by someone other than who designed it. The test case must define the expected output.	Questions to the software that may also be open-ended, even though thinking of the hypothesis is important, documentation need varies.
Test techniques	Preferably mathematical form, enables same results for different users.	Preferably heuristic form, enables useful results for different users.
Level of test-first design	Test design first by tester to find faults in high-level requirements.	Test first design by developer.

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile in Testing (4/6)

Source: Pyhäjärvi, M. & Rautiainen, K. 2004. Integrating Testing and Implementation into Development. Engineering Management Journal, June '04

Aspect	Plan-driven testing	Exploratory testing
Culture	Control, in form of emphasizing entry and exit criteria in testing.	Collaboration, in form of emphasizing testing as service providing value constantly.
Types of processes	Step-by-step processes, requiring as little interpretation as possible.	Backward-forward processes, leaving room for deciding when each process phase should be done.
Role emphasis	Emphasizes test manager and his skills.	Emphasizes each tester and his skills.

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Plan-Driven vs. Agile in Testing (5/6)

Source: Pyhäjärvi, M. & Rautiainen, K. 2004. Integrating Testing and Implementation into Development. Engineering Management Journal, June '04

Aspect	Plan-driven testing	Exploratory testing
Focus of improvement	Focus on developing methods.	Focus on developing skills.
View on practices	Best practices exist and should be applied.	All so-called best practices are just heuristics, rules of thumb.
Pacing testing	Testing to be planned is described as off-synch testing, responding to development in larger increments.	Testing to be planned is mixed off-synch and in-synch, with emphasis on in-synch testing and usually intuition-based describing the off-synch activities.

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

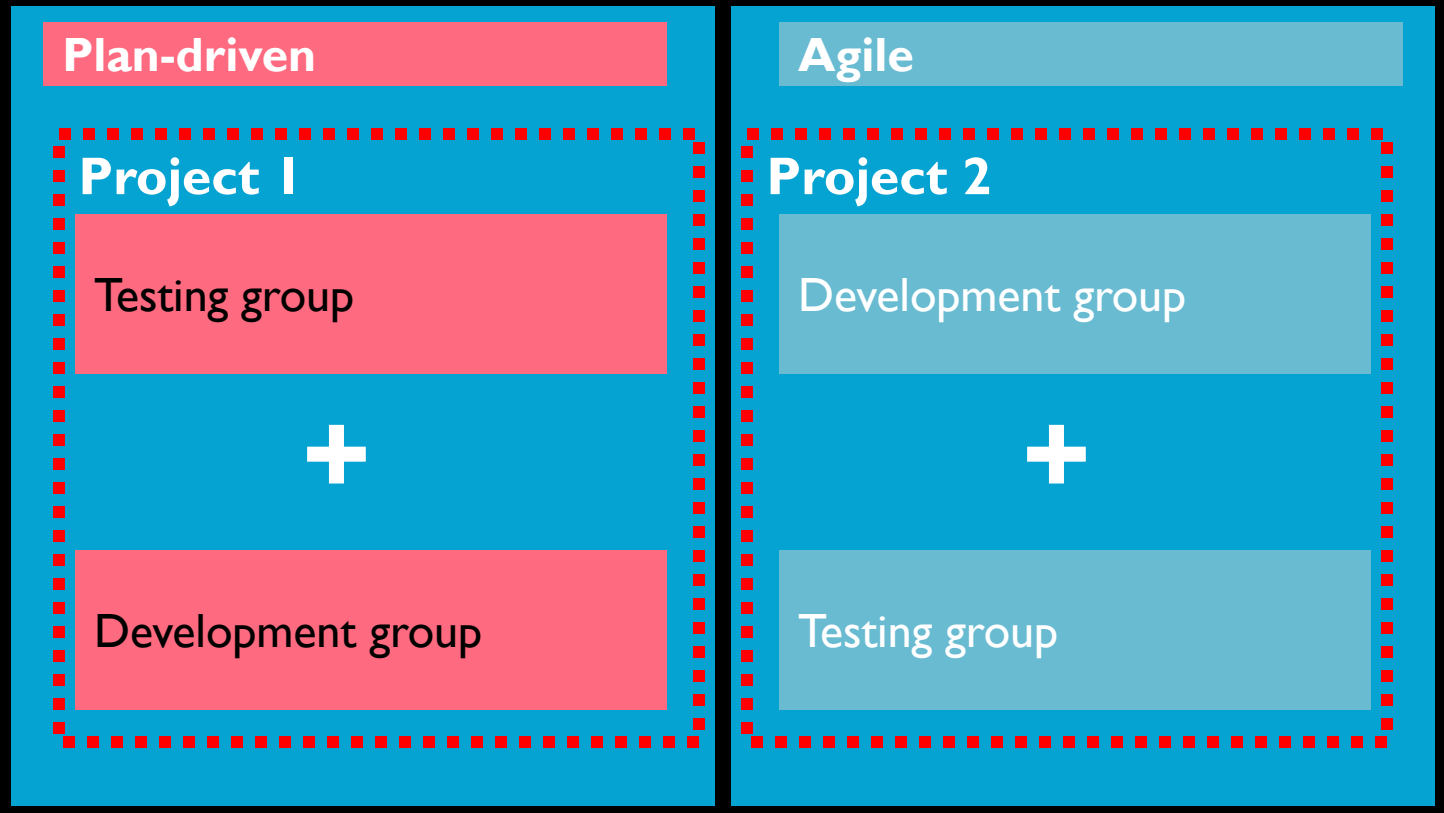
Plan-Driven vs. Agile in Testing (6/6)

Source: Pyhäjärvi, M. & Rautiainen, K. 2004. Integrating Testing and Implementation into Development. Engineering Management Journal, June '04

Aspect	Plan-driven testing	Exploratory testing
Requirements specification	If you don't have a specification, you can't test. Thus you should require unambiguous requirements.	Requirements are useful fiction at best. Documentation is always only fiction – if it's useful, excellent.
Validation and testing	Testing is verification and validation is done if needed.	Validation is an essential part of testing.

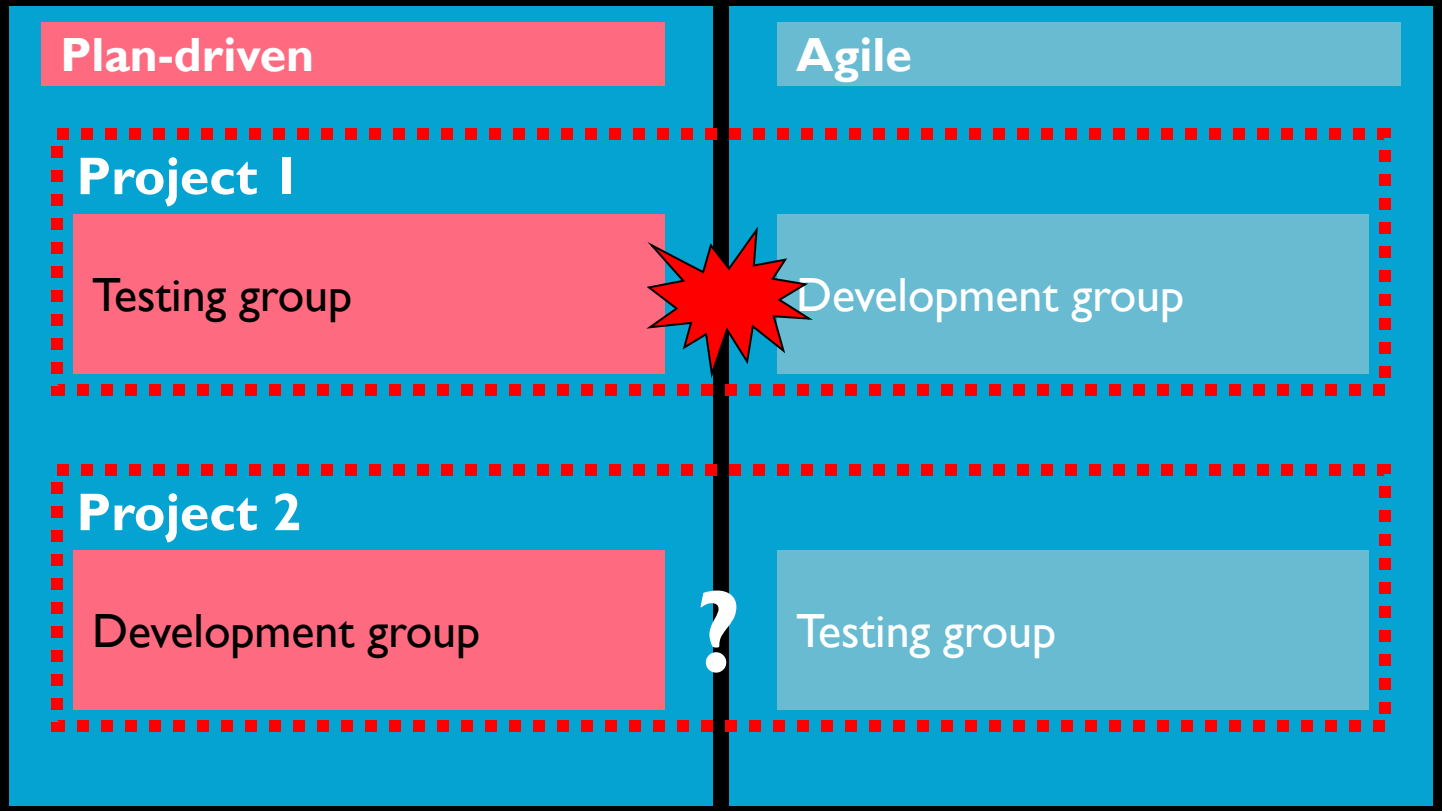
https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Combining Two of a Kind



https://www.linkedin.com/in/maaret/ @maaretp@mas.to

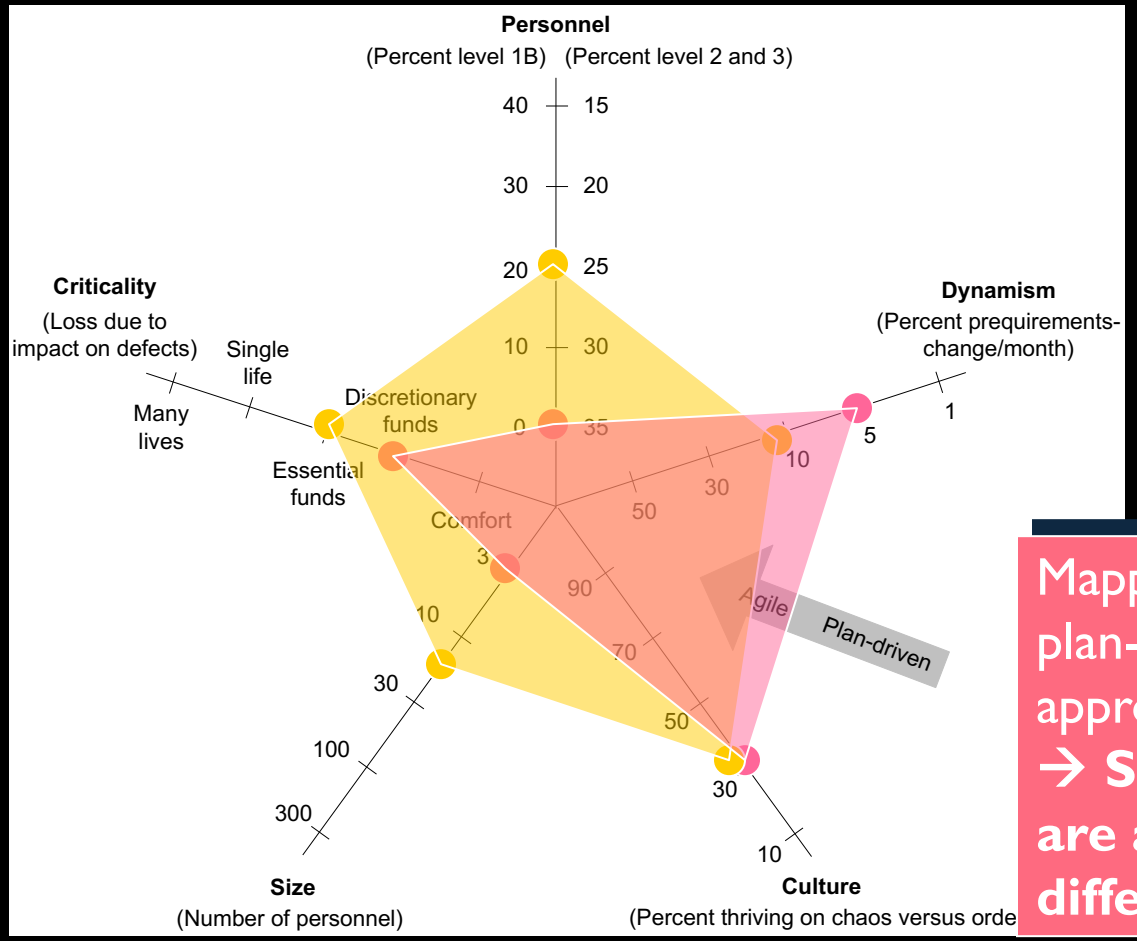
Mix and Match?



https://www.linkedin.com/in/maaret/ @maaretp@mas.to

One Approach to Several Projects Is Sub-Optimal for All

Source: Boehm, B., and R. Turner. 2003. Using Risk to Balance Agile and Plan-driven Methods. IEEE Computer 36, no. 6.



Mapping to analyze need of plan-driven vs. agile approach.
→ **Seemingly similar are actually very different**

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

Conclusions

- Plan-driven and agile methods – including testing – have different home grounds
- Different mindset has its impact on test planning
- Development and testing pieces in the puzzle should fit together
- Getting into the agile mindset has proved a challenge in practice

https://www.linkedin.com/in/maaret/ @maaretp@mas.to