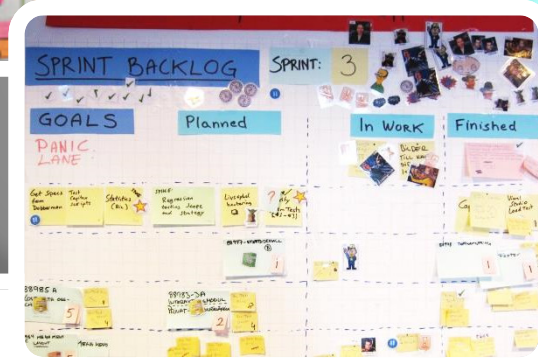
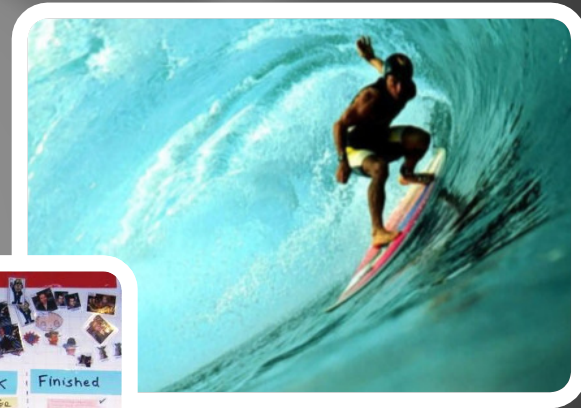
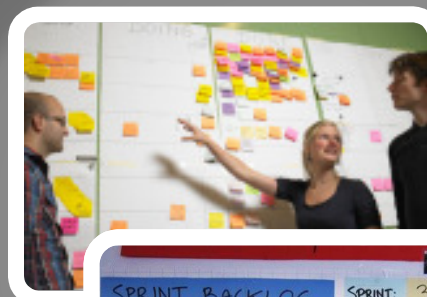


Agile Testing

Does automation replace the tester?

- THE AGILE CONTEXT
- AGILE TESTING
- THE AGILE TESTER
- MYTHS

- EXAMPLES
- TREND SPOTTING



WHO AM I?

Jimmy Janlén

0735 – 12 57 25

jimmy.janlen@crisp.se

www.crisp.se/konsulter/jimmy-janlen



Jimmy
Janlén



Jimmy Janlén

History...

- Developer
- Project Manager
- Scrum Master
- Scrum/Agile Coach
- Teacher
- Team Manager

Work as...

- Agile/Lean Coach
- Scrum Master
- Agile Testing
- Courses
- Seminars

Experience...

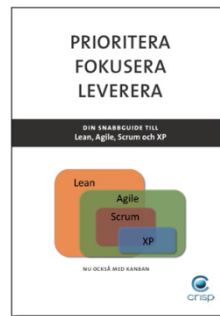
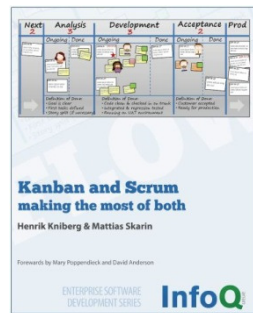
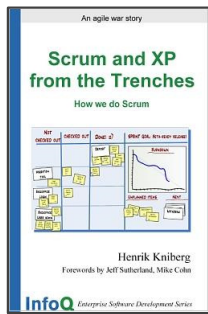




Crisp is an employee owned company known for agile courses with internationally renowned teachers and experienced agile developers and coaches.



Utvecklare User Experience Coacher Lärare



Crisp's Blog
from the Crisp Consultants

Using CloudBees for teaching XP practices | JavaScript Development - A Year Later

Advanced Agile with Alistair Cockburn, Oct 1-2

Subscribe to Henrik's RSS feed

Published on September 10, 2010 by Henrik Kniberg

On Oct 1-2 Alistair Cockburn is in Stockholm teaching Advanced Agile together with me. He has been here twice before and course reviews have been great! Alistair has a very pragmatic down-to-earth style, while maintaining the theoretical depth needed for an advanced course. He's not only a Fun Guy, he's the one who (literally) wrote the book on Agile Software Development, and co-created the agile manifesto.

Join us!



Me @ Online



Crisp
<http://www.crisp.se/konsulter/jimmy-janlen>



@JimmyJanlen



LinkedIn: JimmyJanlen
<http://se.linkedin.com/in/jimmyjanlen>



jimmy.janlen@gmail.com



Swedish Blog
"Den Scrummande Konsulten"
<http://jimmyjanlen.wordpress.com>



Crisp's Blog
<http://blog.crisp.se>



HOME ABOUT

Får du allt av Agile?

Crisps coacher hjälper dig att slå in på vägen av ständig förbättring.



Se alla coacher >

Crisp's Blog

from the Crisp Consultants

Role Expectation Mapping

Posted on March 11, 2014 – 1:44 am by Jimmy Janlén

Role Expectation Mapping is a series of workshop that explores, clarifies and establishes which expectations members of a group, team or project have on each other.

If you suspect that collaboration is undermined because of mismatch of expectations between people, then this exercise could boost the team's ability to collaborate efficiently together. It is also a powerful way to jump start a new team and give them a structure to relate to.



Search

Subscribe via RSS

Guest Contributors



Christopher Avery



Ellen Gottesdiener



Jeff Gothelf

Blog Authors



Thomas Thyberg



Evil Coach



Jimmy Janlén

Me @ Online



Crisp Agile Academy

<https://www.youtube.com/crispagileacademy>



Jimmy Janlén

“Agile Testing is only a
meaningful in an agile context”

“Agile Testing is only a meaningful in an agile context”



“Agile Testing is only a meaningful in an agile context”

Agile
Acceptance
Testing?

Agile
Test Team?

Agile
Test Tools?

Agile
Tester?

Agile Test
Reports?



A photograph of a surfer riding a barrel wave. The surfer is in a crouched position, wearing a light-colored long-sleeved shirt and dark shorts. The wave is a deep blue color and is curling over the surfer, creating a tunnel effect. The background is a clear blue sky.

Agile Testing =

Testing that enables
AGILITY

“Agile Testing is only a meaningful in an agile context”



Agile is...

Working & Valuable Products.

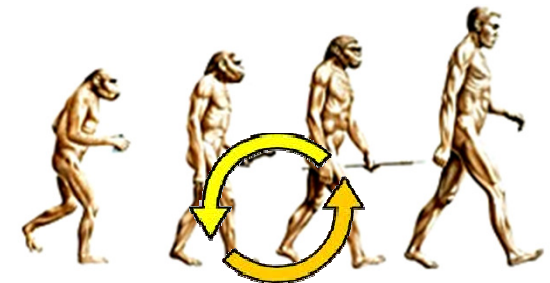
Deliver business value early & frequently.
Incremental/Continuous delivery.
Short iterations or flow.



Welcoming change.
Managing change.
Respond to change.

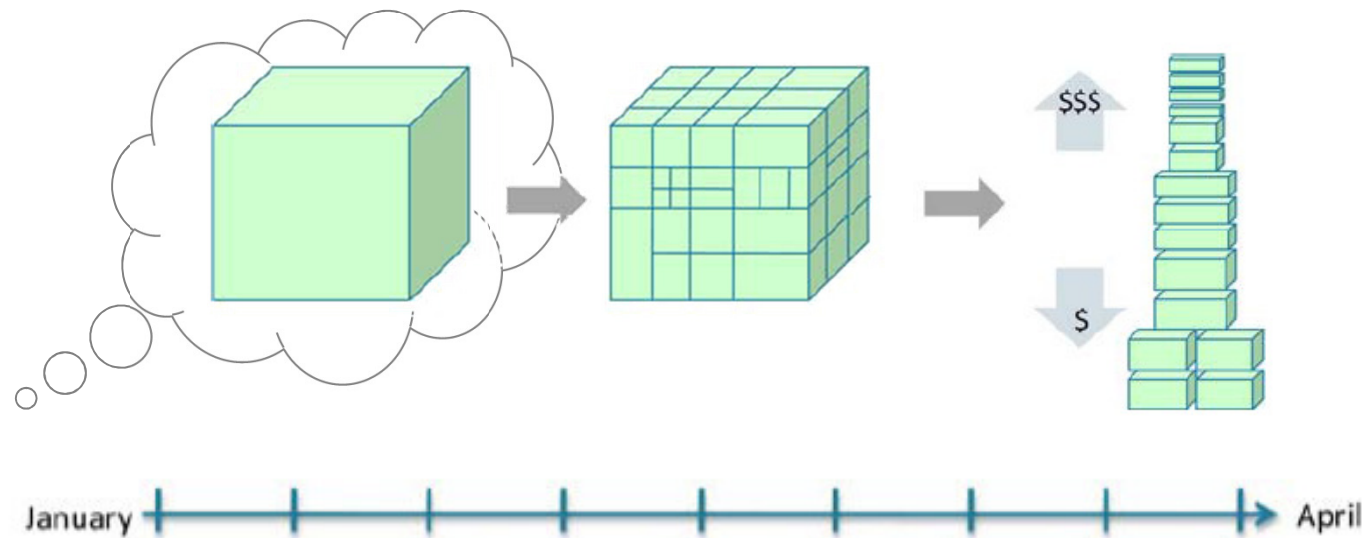


Customer collaboration.
Close and continuous.
Face-to-face.



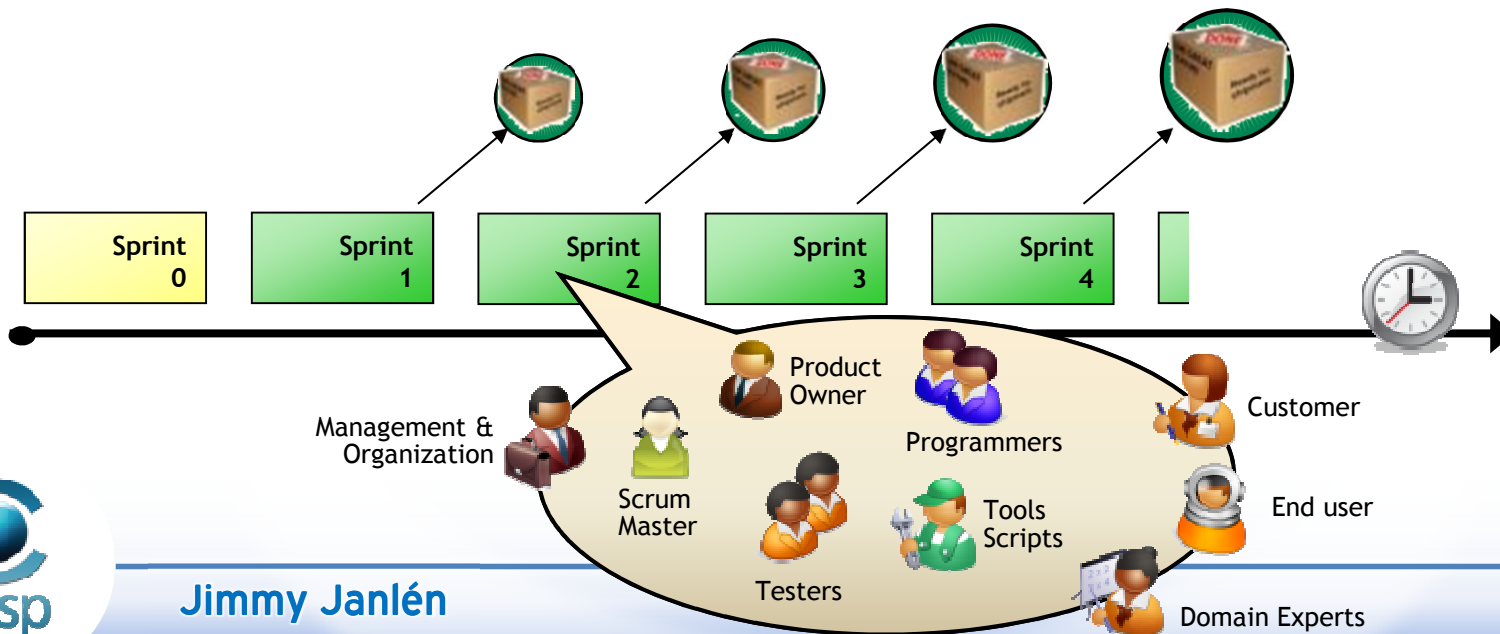
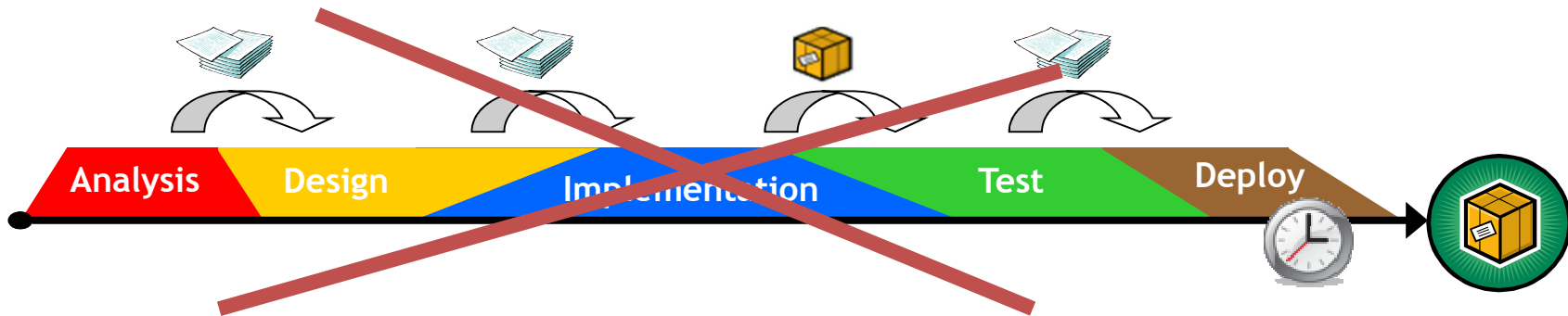
Continuous learning.
Self-adapting process.
Transparency.

Iterative incremental development



- Unit tested
- Integration tested
- Regression tested
- Performance tested
- Acceptance tested
- Useful & Valuable

Incremental delivery and short iterations

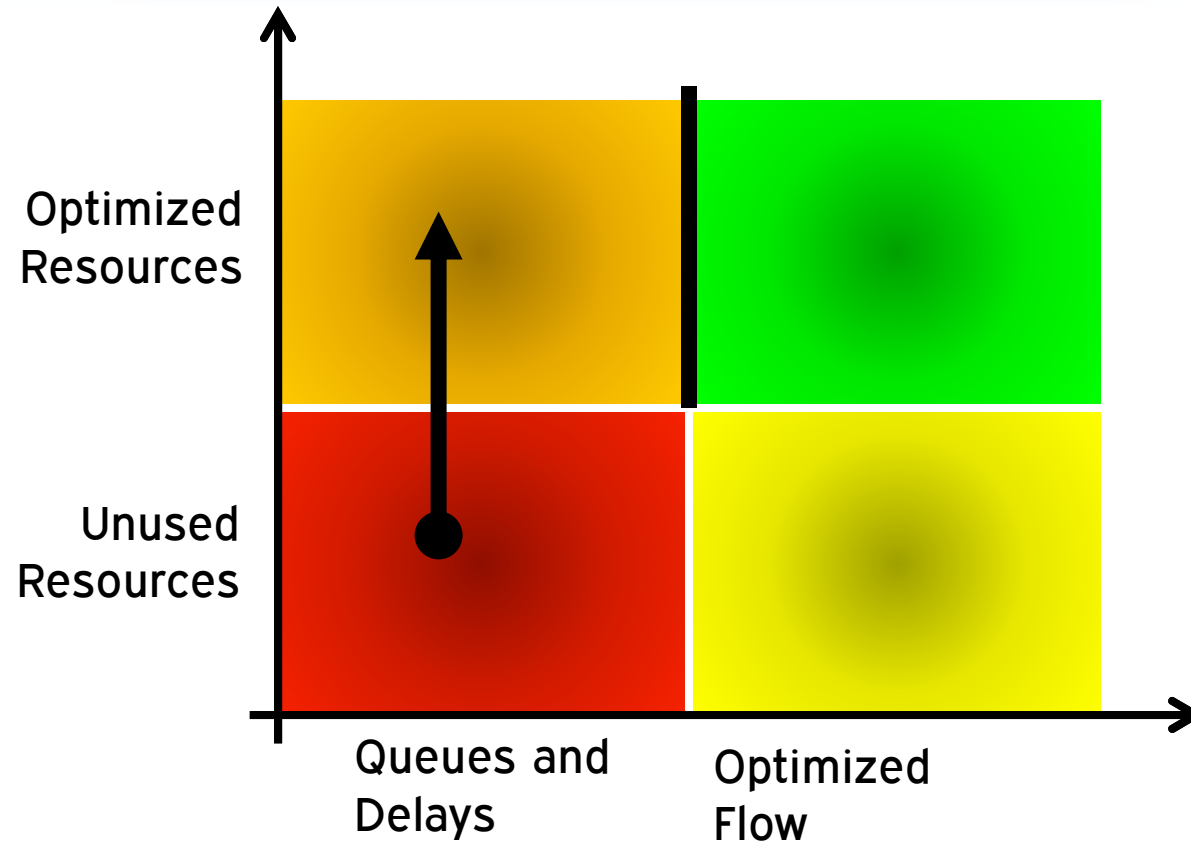


Lean - Summarized and Simplified

- Maximize customer value
- Minimize delivery time
- Minimize waste

**Continuously
improve (kaizen)**

Lean - Summarized and Simplified



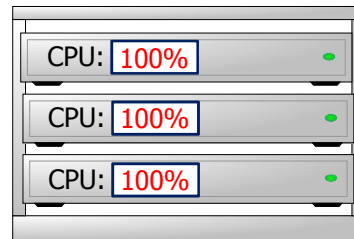
Lean - Summarized and Simplified



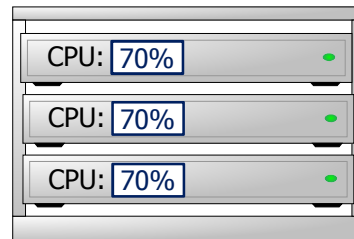
Road utilization = 100%
Throughput: Terrible



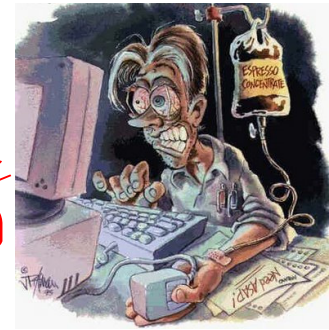
Road utilization \approx 70%
Throughput: Fast



Server utilization = 100%
Response: Slow



Server utilization = 70%
Response: Fast

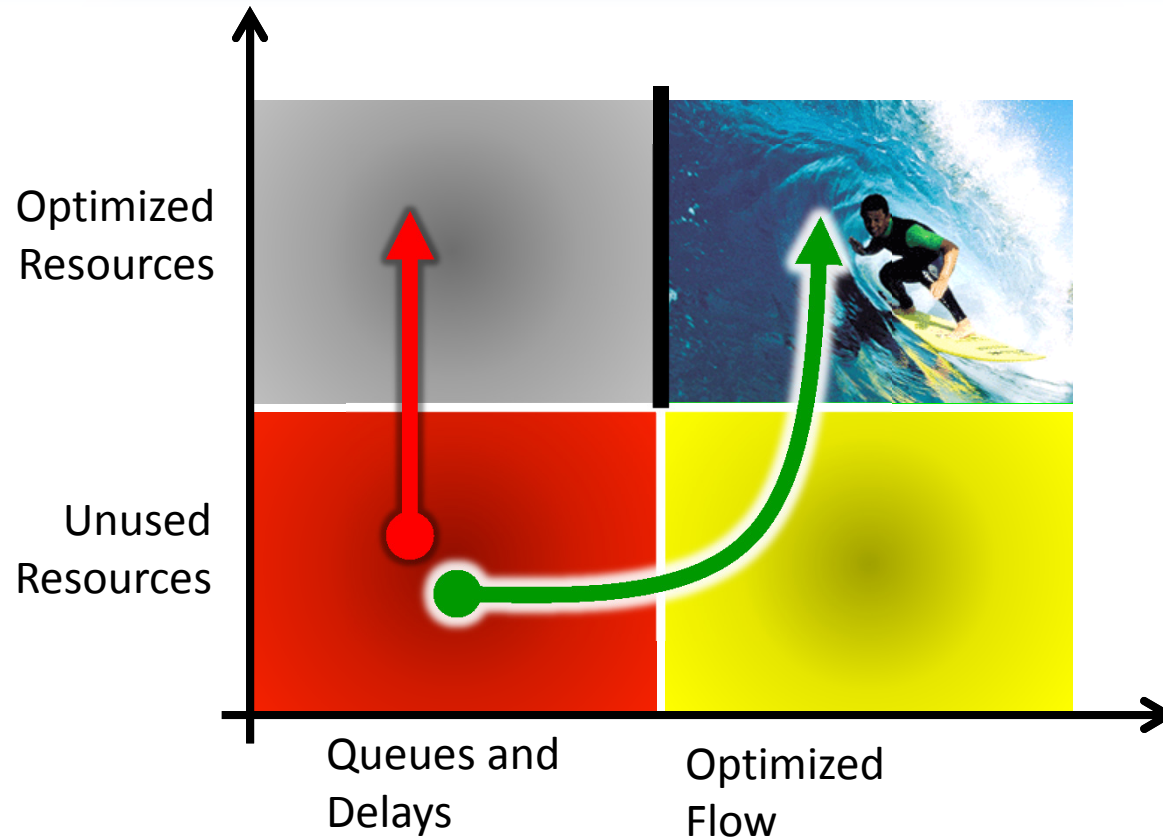


Staff utilization = 100%
Delivery: Slow



Staff utilization \approx 70%
Delivery: Fast

Lean - Summarized and Simplified



The seven wastes of software development

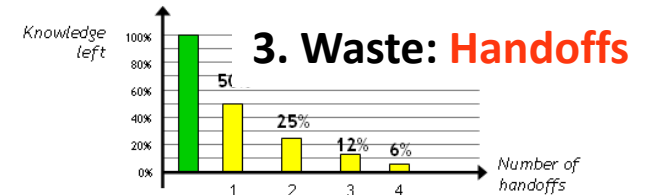
1. Waste: Partially Done Work

Examples:

- Uncoded documentation
- Unsynchronized code
- Untested code
- Undocumented code
- Undeployed code

2. Waste: Extra features

Don't predict nor guess!

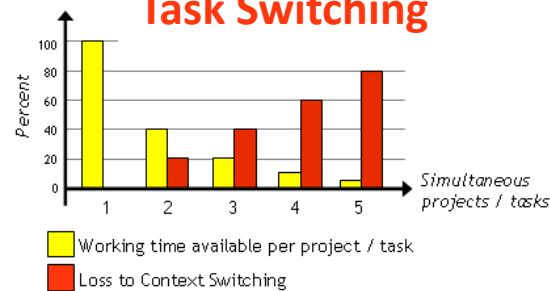


4. Waste: Relearning

Examples:

- Poor knowledge sharing
- Poor documentation
- Repeating mistakes

5. Waste: Task Switching



6. Waste: Delays and Motions



7. Waste: Defects



The seven wastes of software development

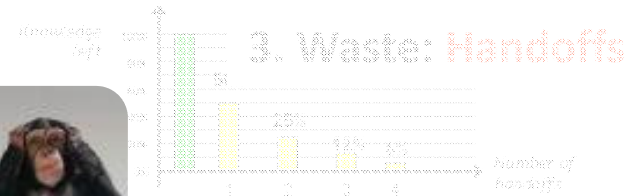
1. Waste: **Wishful Thinking**

Examples:

- * Uncoded documentation
- * Unsynchronized code
- * Untested code
- * Undocumented code
- * Undeployed code



Waste: **Fear**



4. Waste: **Relearning**

Examples:

- * Poor knowledge sharing
- * Poor documentation
- * Repeating mistakes

5. Waste: **Task**



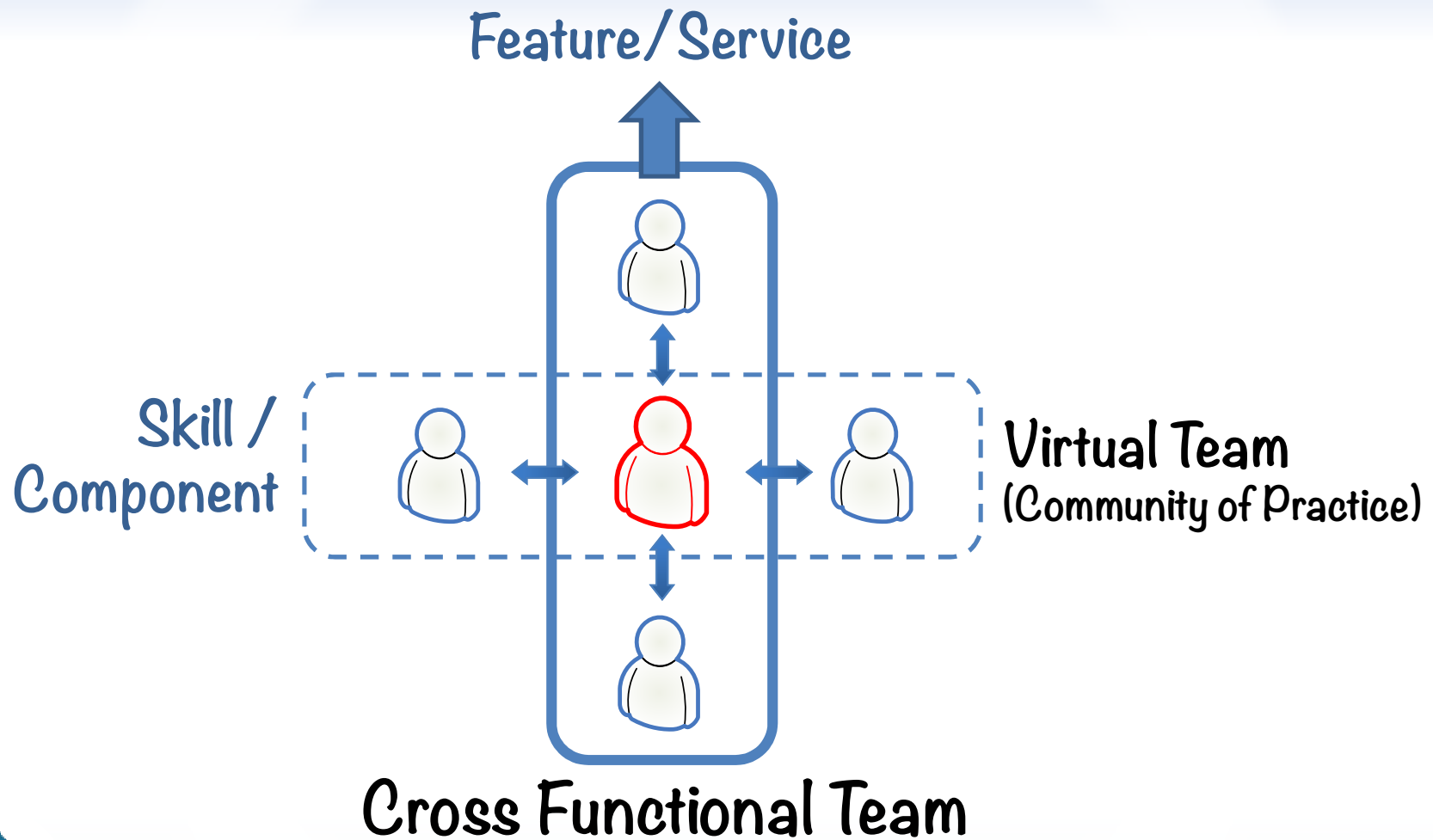
6. Waste: **Delays and Motions**



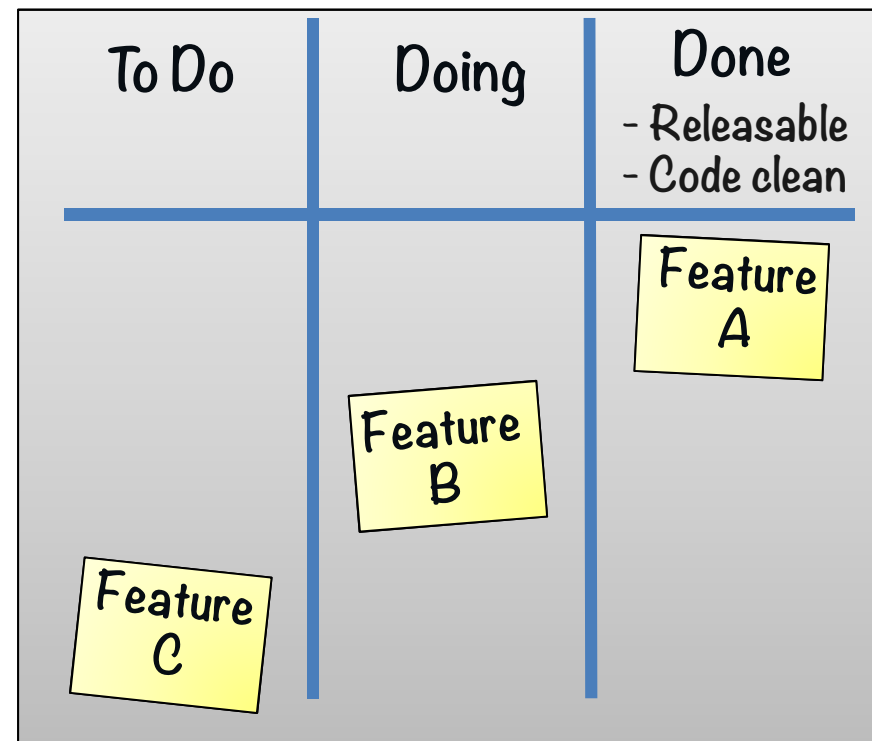
7. Waste: **Defects**



Organize by Feature/Service!



Design, build, test and ship together! One feature at a time



Agile Testing

Why the confusion?

Agile Manifesto

www.agilemanifesto.org

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over **processes and tools**

Individer och interaktioner framför processer och verktyg

Working software over **comprehensive documentation**

Fungerande programvara framför omfattande dokumentation

Customer collaboration over **contract negotiation**

Kundsamarbete framför kontraktsförhandling

Responding to change over **following a plan**

Anpassning till förändring framför att följa en plan

That is, while there is value in the items on the right, we value the items on the left more.

Principles behind the Agile Manifesto

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



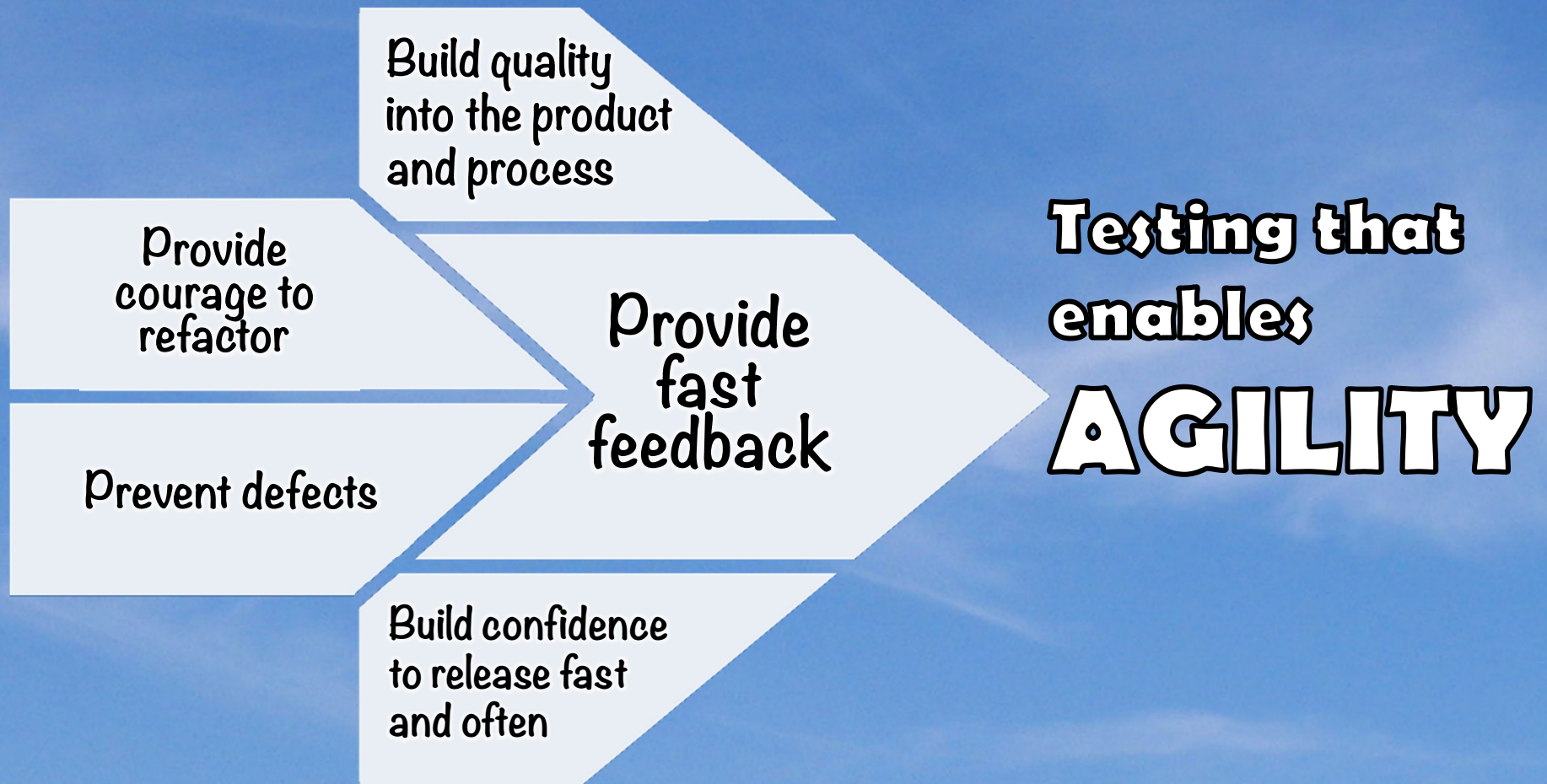
David Evans

"Testing slows down development just as passengers slows down the bus.

The speed of the bus is not the point!"



Purpose of Tests & Testing



Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ **Uses lightweight test techniques**
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

Characteristics of an agile test approach...



Adheres and obeys to the principles of Agile & Lean...

- ✓ Embraces changes late in development and manages uncertainty in upcoming features
- ✓ Handles and utilizes that testing and development happens at the same time in close collaboration
- ✓ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ✓ Uses lightweight test techniques
- ✓ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ✓ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ✓ Introduces demands on test environment and support processes
- ✓ Is continuously challenged and improved

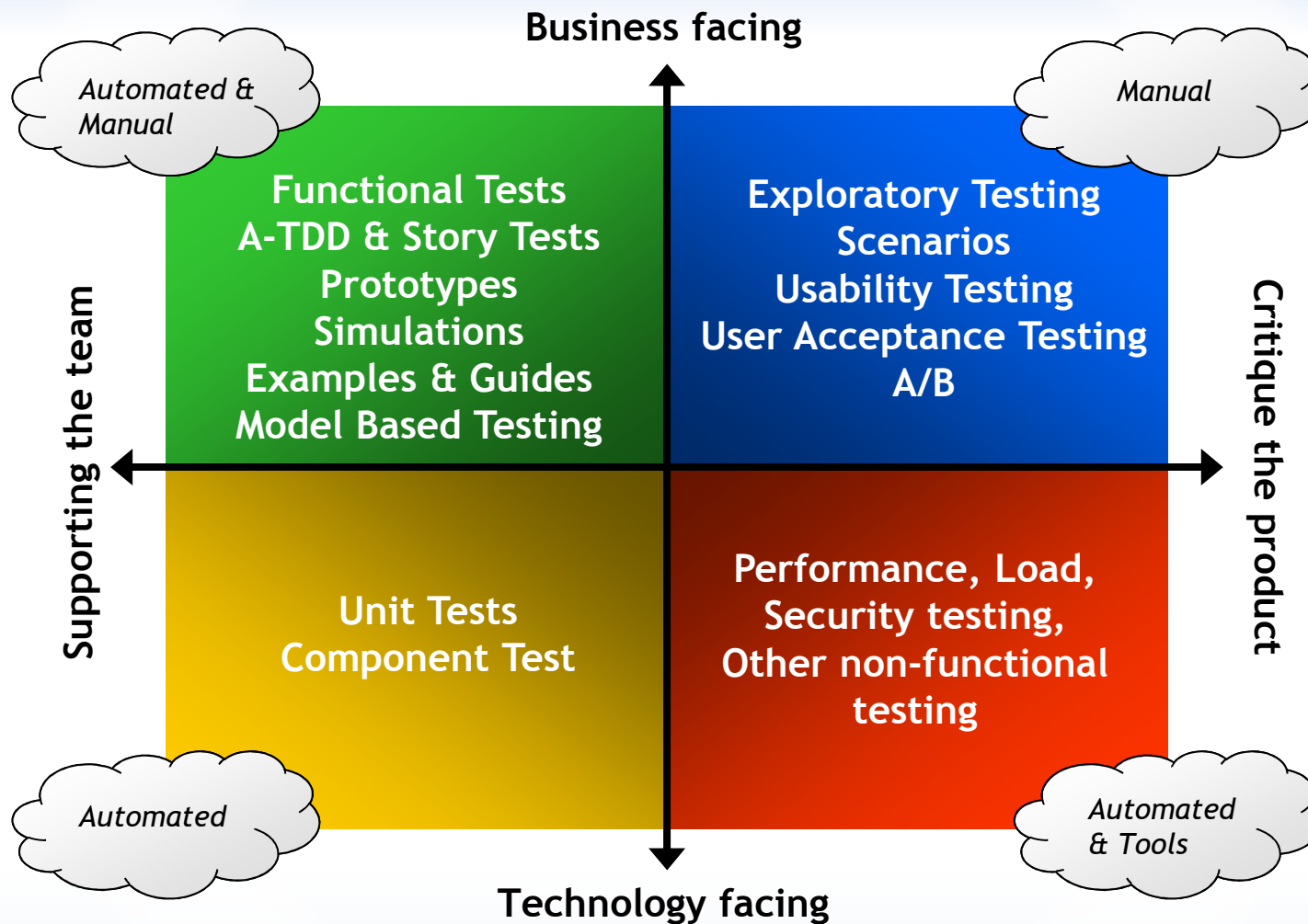
Characteristics of an agile test approach...

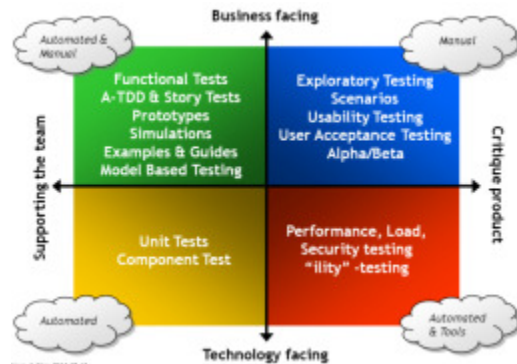


Adheres and obeys to the principles of Agile & Lean...

- ☑ Embraces changes late in development and manages uncertainty in upcoming features
- ☑ Handles and utilizes that testing and development happens at the same time in close collaboration
- ☑ Fits within a sprint and aims at a potential release (that is ready to be delivered)
- ☑ Uses lightweight test techniques
- ☑ Incorporates and utilizes a cross-functional team, i.e. that the skills and experts are in the team
- ☑ Works with a test base (test data, test suits, documentation, etc.) that grows and evolves over time
- ☑ Introduces demands on test environment and support processes
- ☑ Is continuously challenged and improved

Agile Test Quadrants





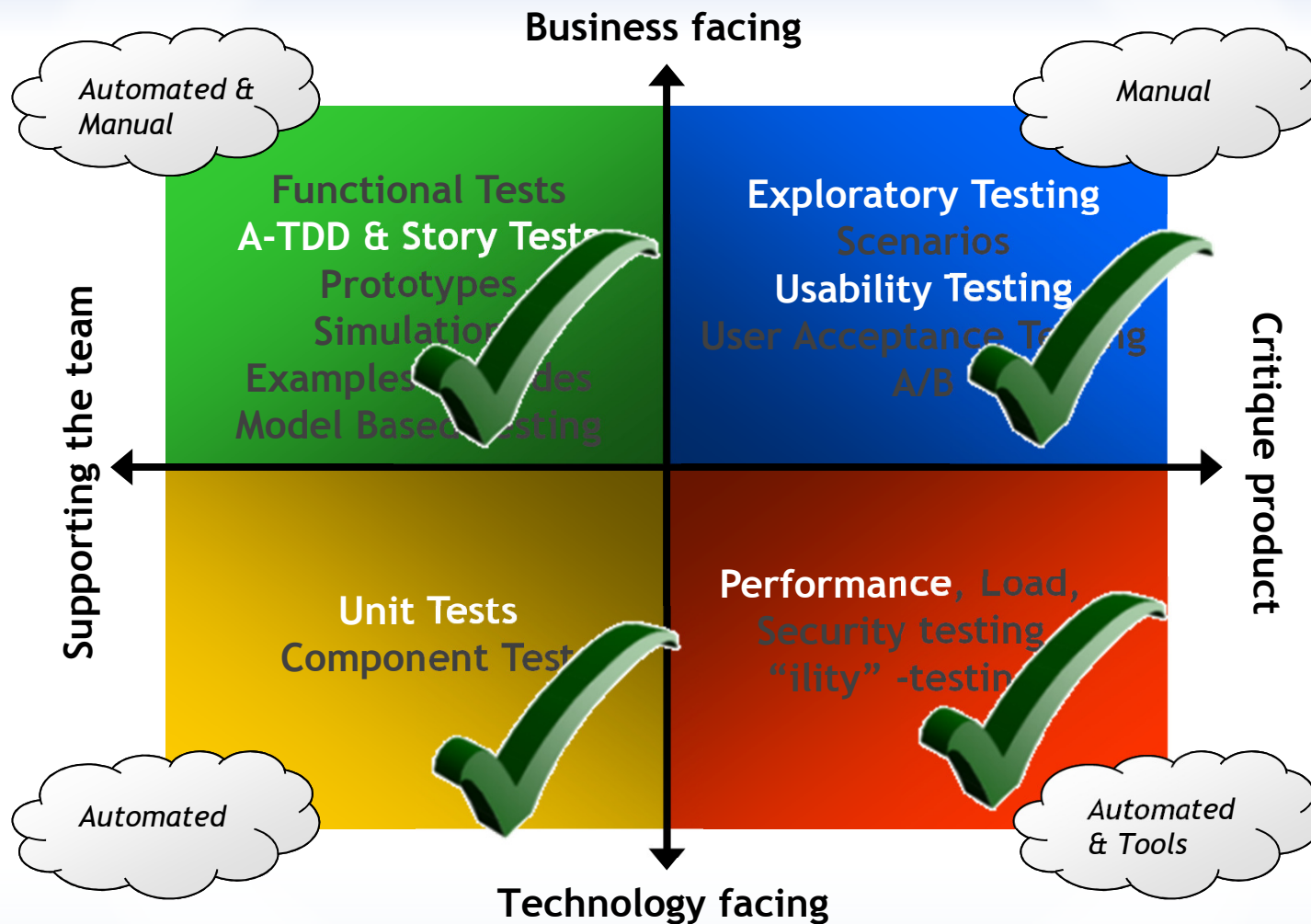
**How can we do
ALL of this – EACH Sprint?**

Decrease scope



KISS – Keep It Simple Stupid

Agile Test Quadrants



Being DONE
VS
Achieving DONE

Being DONE vs Achieving DONE



- DONE/Story**
- Designed
 - Coded
 - Versioned
 - Tested
 - Documented
 - Deployable
 - Approved

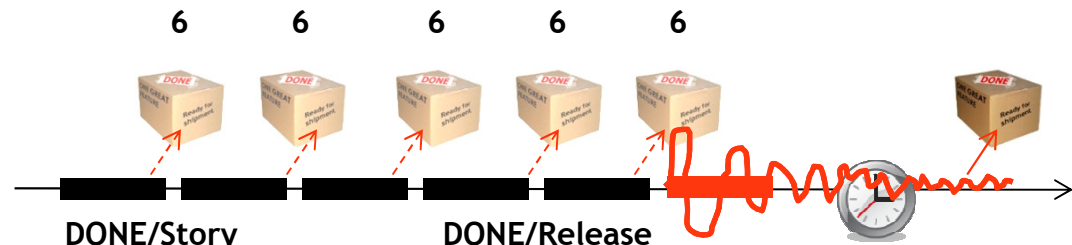
- DONE/Release**
- ✓ Already fulfilled



Definition Of DONE

DONE

- Designed
- Coded
- Versioned
- Tested
- Documented
- Deployable
- Approved



- DONE/Story**
- Designed
 - Coded
 - Versioned
 - Tested

- DONE/Release**
- Tested
 - Documented
 - Deployable
 - Approved

DoD examples

You could benefit from different Definition of DONE for...

Definition of
DONE for
User Stories

Definition of
DONE for
Releases

Definition of
DONE for
Bug Fixes

Definition of
Ready

Definition of DONE User Stories



Designed

- Clear User Story (INVEST)
- Additional Acceptance Criteria specified if necessary (in addition to DONE/User Story)
- UI Mock-up / Prototyped (if applicable)
- Acceptance tests written (end-to-end, i.e. both functional, non-functional and integration)
- Design/Solution is auto testable ?*
- Consider:
 - Licenses ?
 - Bigger architecture ?
 - Wider ui design and user experience ?

Tested

- Acceptance Tests Automated (when possible) or Acceptance tests executed
- Extend performance tests (if necessary)
- Unit Tests added to C.I.

Coded

- Code written using TDD & Unit Tests
- Documented on Wiki (if relevant)
 - Examples: Architecture overview, State charts, Example, Guidelines, Protocols, XML-formats, etc.
- Prepared for localization
- Versioned
 - Code checked into GIT with reference to JIRA Issue

Documented

- External documentation updated (outside R&D):
 - Branding Guidelines
 - Installation and upgrade instructions
 - User Manuals
 - Product Information
 - Release Notes
 - Customer Licenses
 - Performance
- Internal documentation:
 - Internal R&D Licenses

Deployable

- Release Build and Packaged

Approved

- Feedback on UI Mock-up / Prototype (when applicable)
- Sprint Demo prepared (when demonstratable)



The Agile Tester

Attitude and Mindset

"The job of tests, and the people that develop and run tests, is to prevent defects, not to find them."



Mary Poppendieck

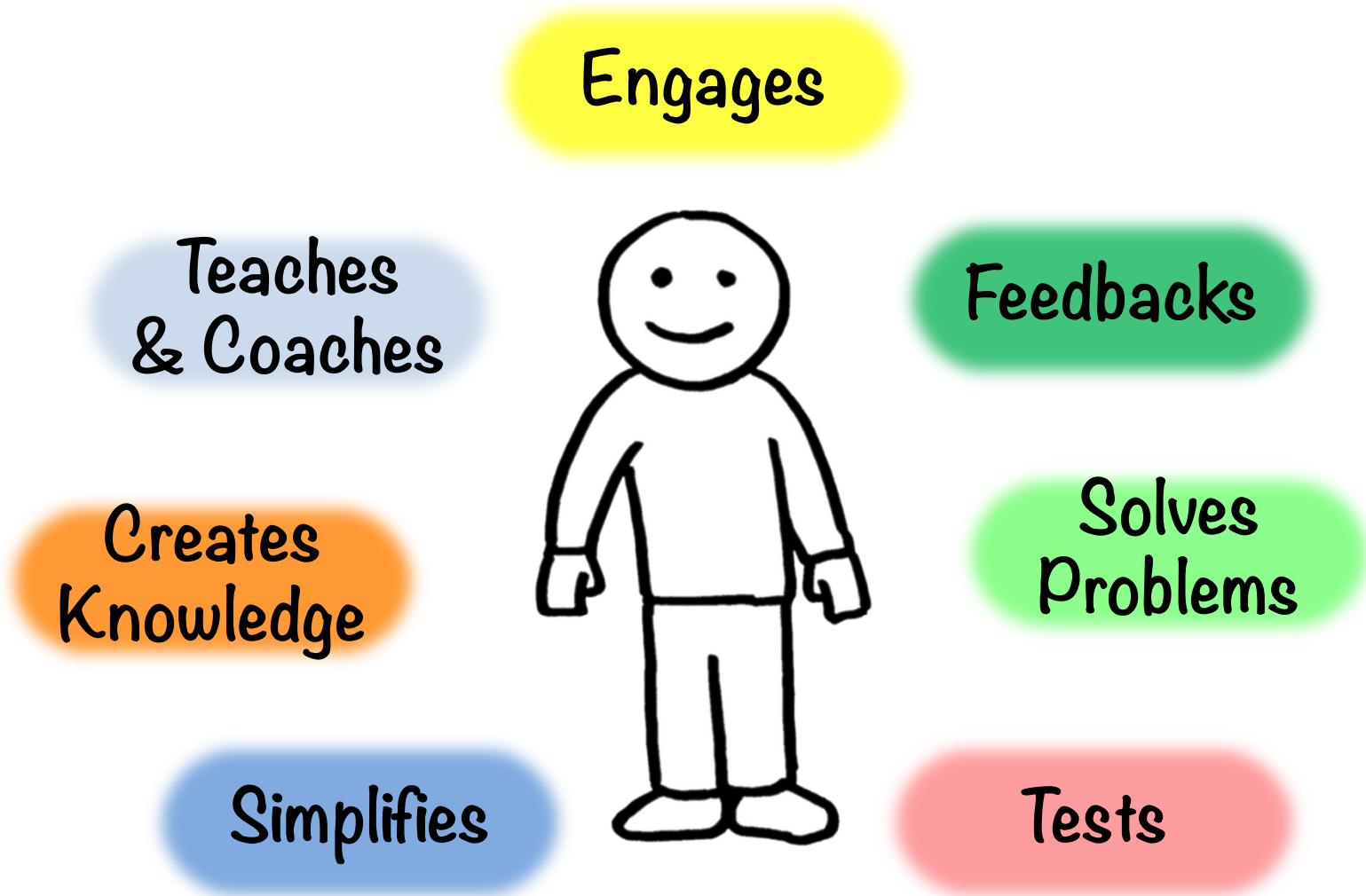
The Agile Tester: Chief of Safety & Guardian of Quality!



If we were out adventuring...



What does an agile tester do?



Myths on Agile & Testing

Myth: It's harder to estimate testing when doing Scrum

BUSTED

IT'S EASIER!

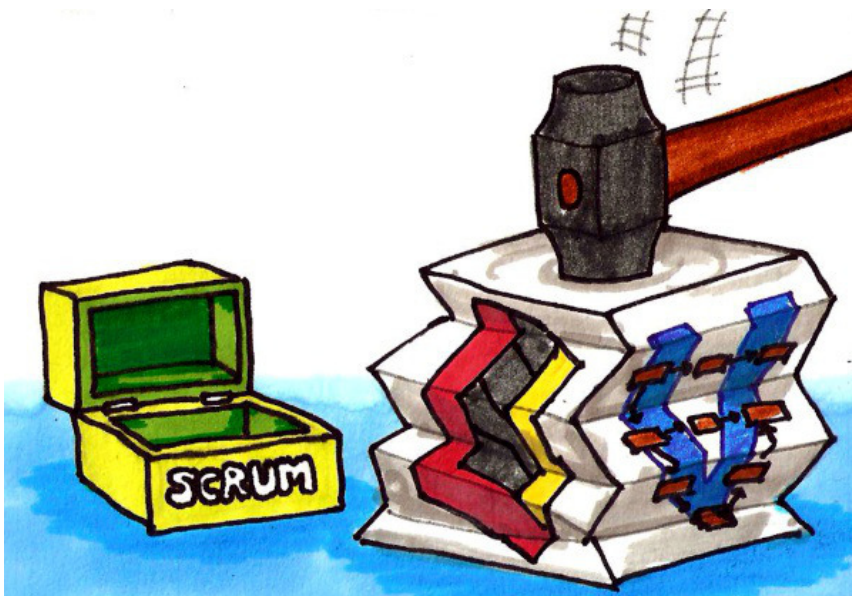
- Scope is smaller = Less to estimate
- Smaller deliverables, less risk
- Testers part of planning and design discussions
- Short learning cycles, estimations get better every sprint



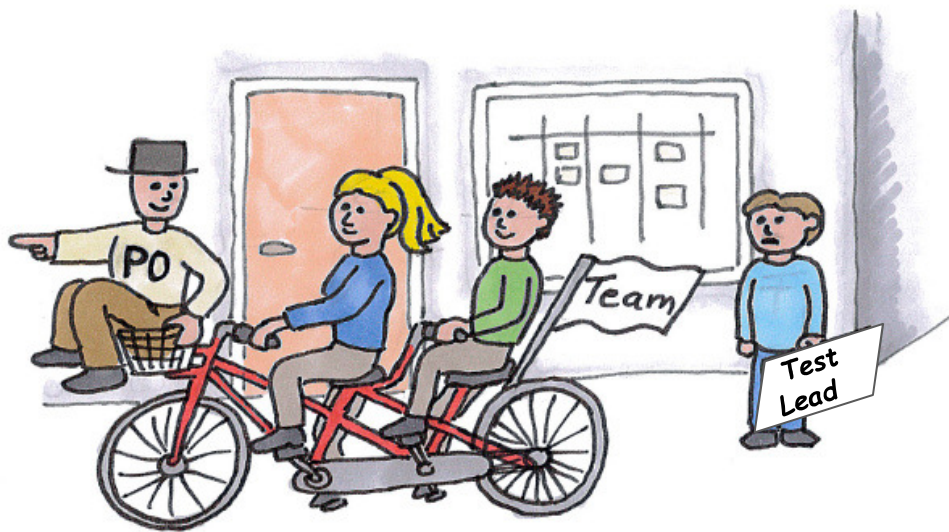
Myth: No time for regression tests when doing Scrum

CONFIRMED

- Impossible to shrink old “non agile” test process to fit into Agile/Sprints.
- Traditional manual regression testing is done after code/feature freeze. Not enough time to test “everything” every sprint.



Myth: An agile team doesn't have a Test Lead



CONFIRMED

IF WE HAD ONE IT WOULD...

- Undermine collective ownership of quality, test process and testing
- Signal lack of trust in skills
- Signal mistrust in motivation
- Only one person doing the thinking = less learning

<http://agilamyter.wordpress.com>



Agila Myter BUSTED (och några bekräftade)

[Hem](#) [Om bloggen](#) [Om oss](#)

Om oss



Jagannath "Jagge" Tammeleht
Agil testare, Scrum Master, testledare. föreläsare och thought leader på **Claremont AB** (konsult). Co-driver av **testzonen**.



Jimmy Janlén
Agile/Lean coach och lärare på **Crisp** (konsult). **Bloggar**, håller seminarier och är intresserad av **grafisk dokumentation** och continuous discovery.



Alexander Tarnowski
Utvecklare, arkitekt och byggmiljökonstnär på **Crisp** (konsult). Skriver på en bok, **"Developer Testing"**.



Therése Ressel
Agil testare, testledare och Scrum Master på Kambi Sports Solutions. Skribent för **testzonen**.

"Myter är som kamouflerad kollektiv visdom, det finns alltid något att lära. Ingen rök utan eld."



SÖK

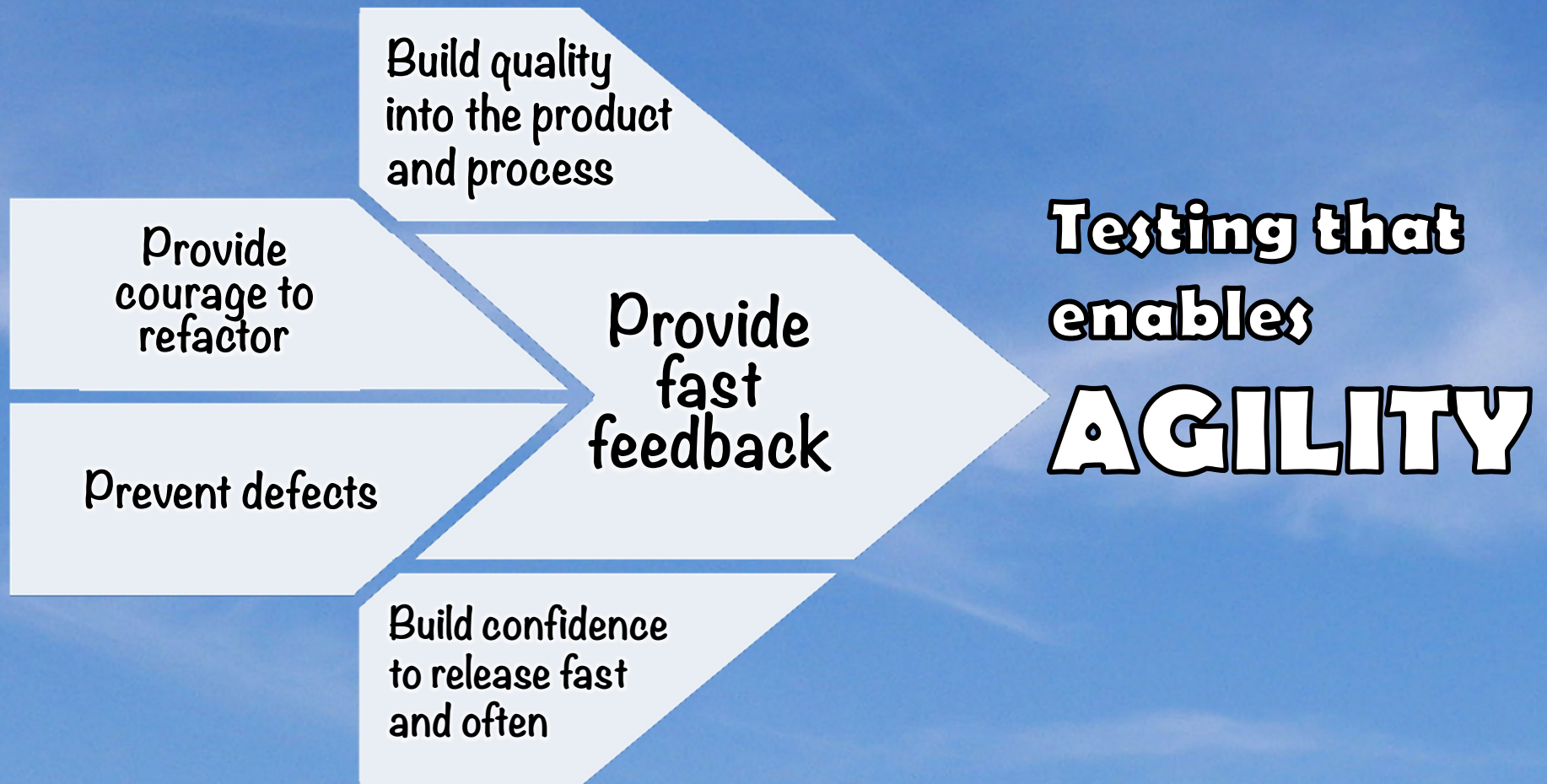
SENASTE MYTER

- > Myt #3: Agila team behöver inga testare
- > Myt #2: Regressionstest hinns inte med
- > Myt #1: Alla agila testare måste vara tekniska

ÄLDRE MYTER OM...

- > Agil Testning

Purpose of Tests & Testing



Examples of Agile Test Strategies & Approaches



From 2 releases a year
to every sprint

Long Stabilization/Test phase



25 people.
1 Back-end team. 1 Front-end team.
1 product.



“Poor respect/care for quality”



Huge defect-list legacy
(avg. age ~ 100 days)



Long time between releases (~2/year)



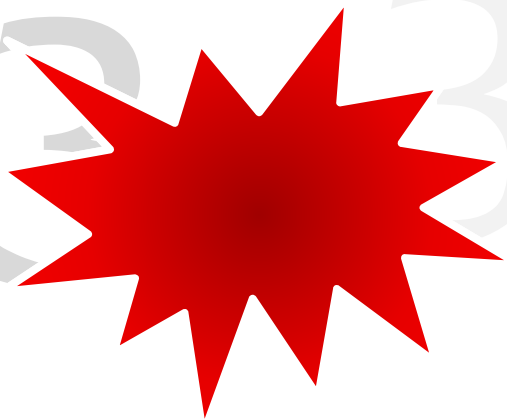
Delays opened up for Scoop creep...





ZERO BUG POLICY





RELEASE EVERY SPRINT!

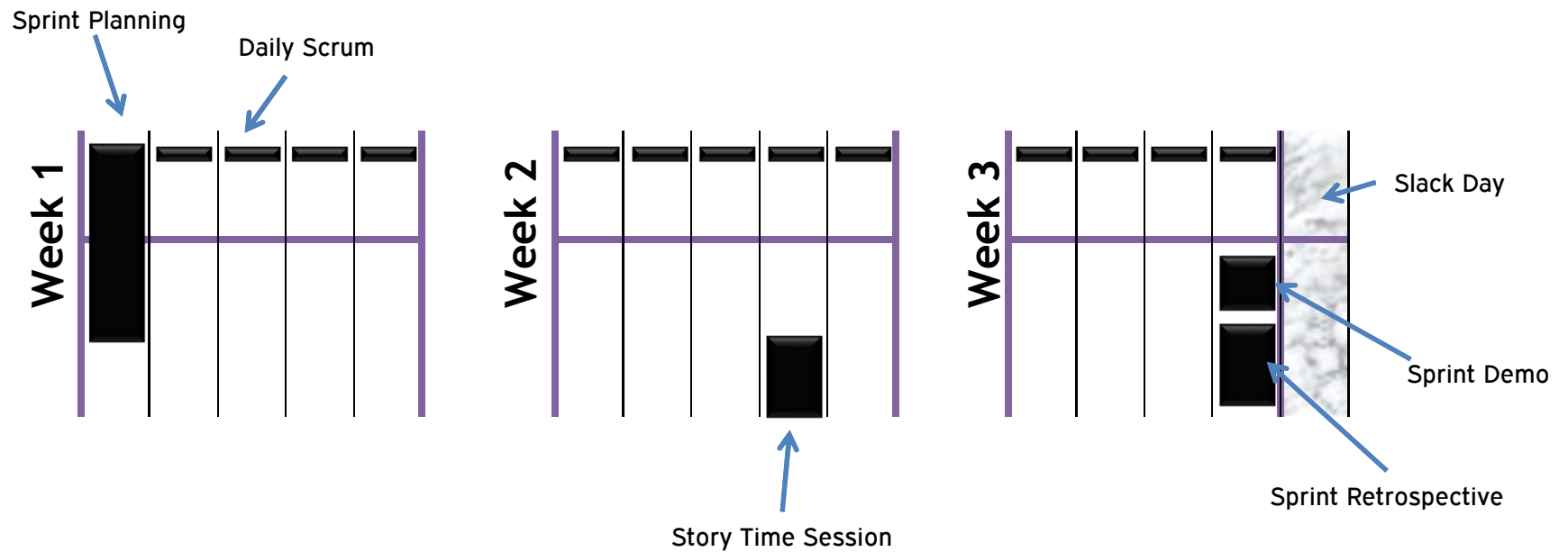




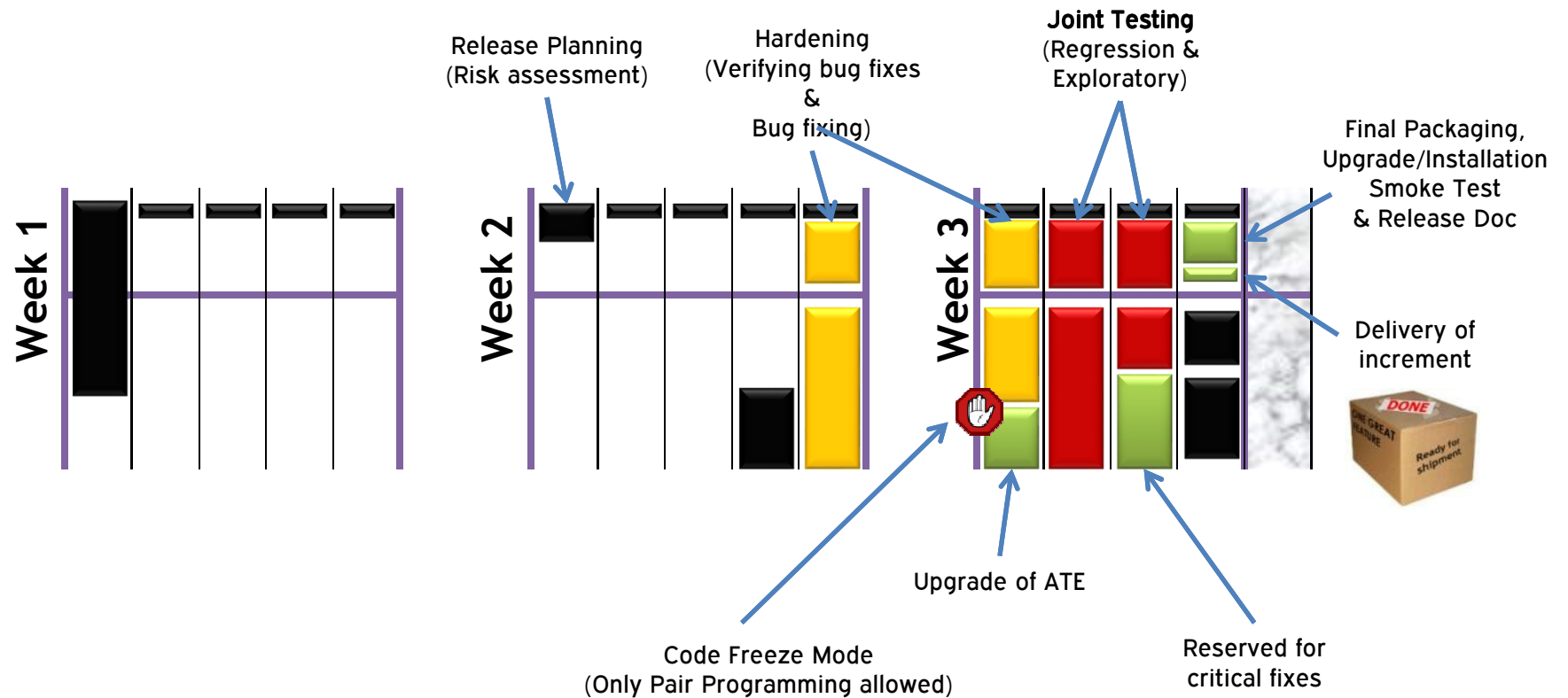
3 XFT TEAMS



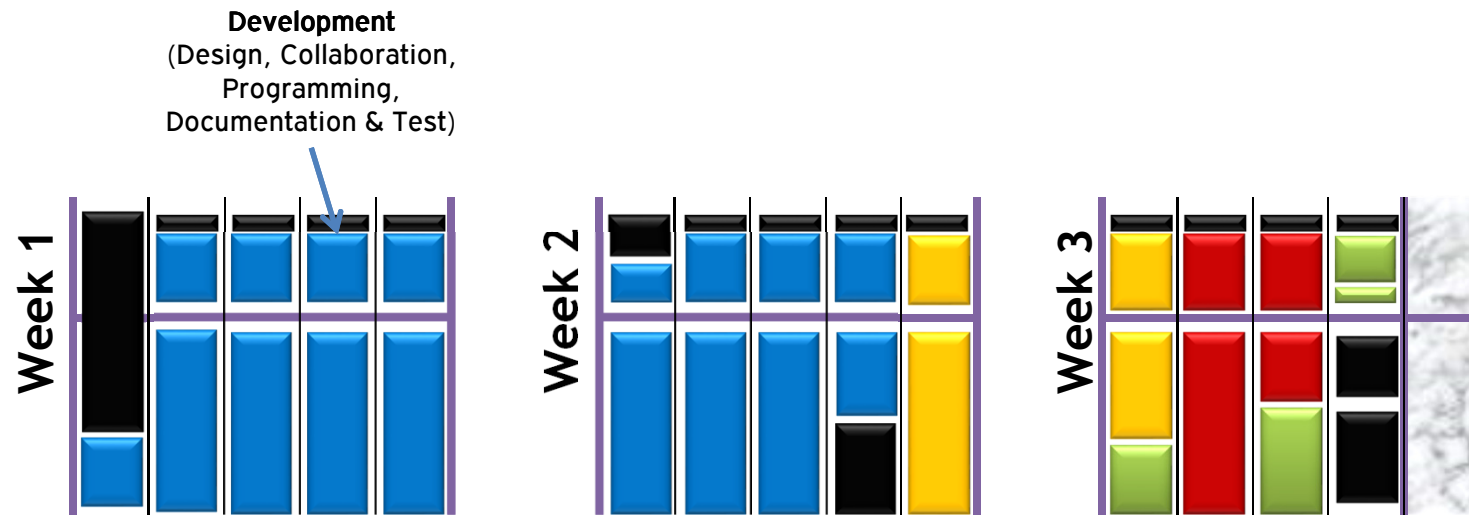
Sprint Test & Release Strategy Example



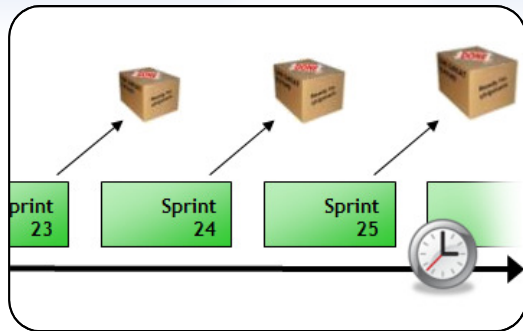
Sprint Test & Release Strategy Example



Sprint Test & Release Strategy Example



Strategy



Release every Sprint!
Rigid Definition of
DONE.



Rotating
Responsibility



Joint Testing



Bug alarm

**6 MONTHS
LATER**



“Higher respect/care for quality”

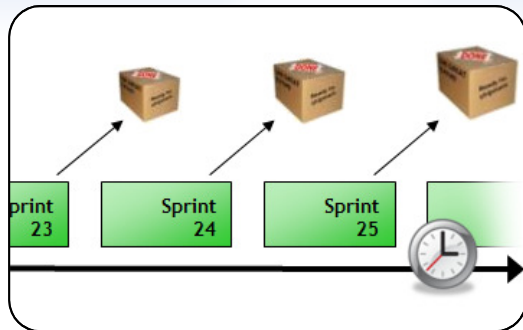


Less Defect-legacy
(avg. age 30 days)



More frequent releases
(4-6/year)

Strategy



Release every Sprint!
Rigid Definition of
DONE.



Rotating
Responsibility



Joint Testing



Bug alarm

TODAY



“Higher respect/care for quality”



Less Defect-legacy
NO DEFECT PAIN!



More frequent releases
EVERY SPRINT! (Every 3rd week)

SIEMENS

medical

Large Scale Scrum

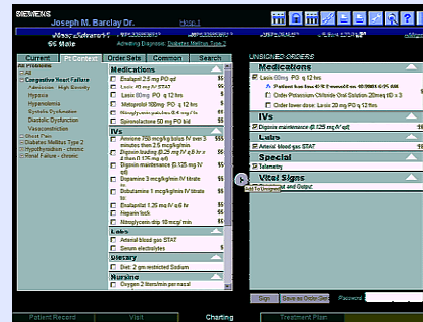
Many customers
 Many wills
 One application



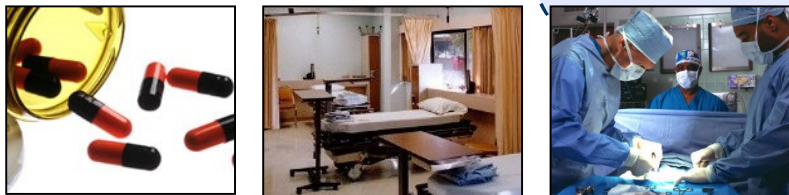
Many customer



Different markets



Soarian®



Complex needs



Changing legal demands

Implemented Quality mechanisms and Disciplines



- ↳ Obeying DONE
- ↳ Acceptance Test Driven Development
 - Clear User Stories (incl. How to demo)
 - PO wrote Acceptance Test drafts in Excel
 - Given... When... Then... / Tables / Etc.
 - Fitness
- ↳ Test Driven Development
- ↳ Automated integration tests
- ↳ Full Nightly Builds
 - + STOP-THE-LINE
- ↳ Team Micro Builds (Compilation & Automated tests)
- ↳ Team Design Review
- ↳ Pair programming, Pair Testing
- ↳ Exploratory Testing
- ↳ Full day joint test efforts
- ↳ Root Cause Analysis



Spotify®

End-to-end responsibility

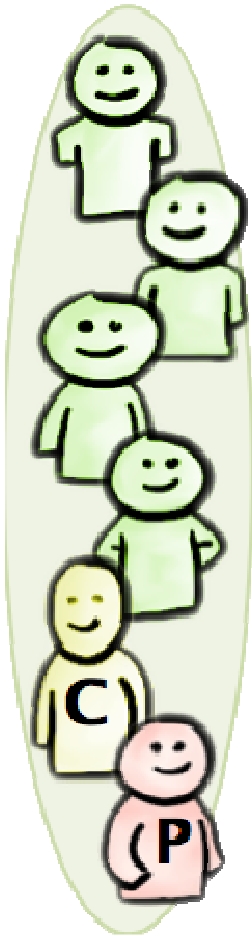
Autonomy



Example squads



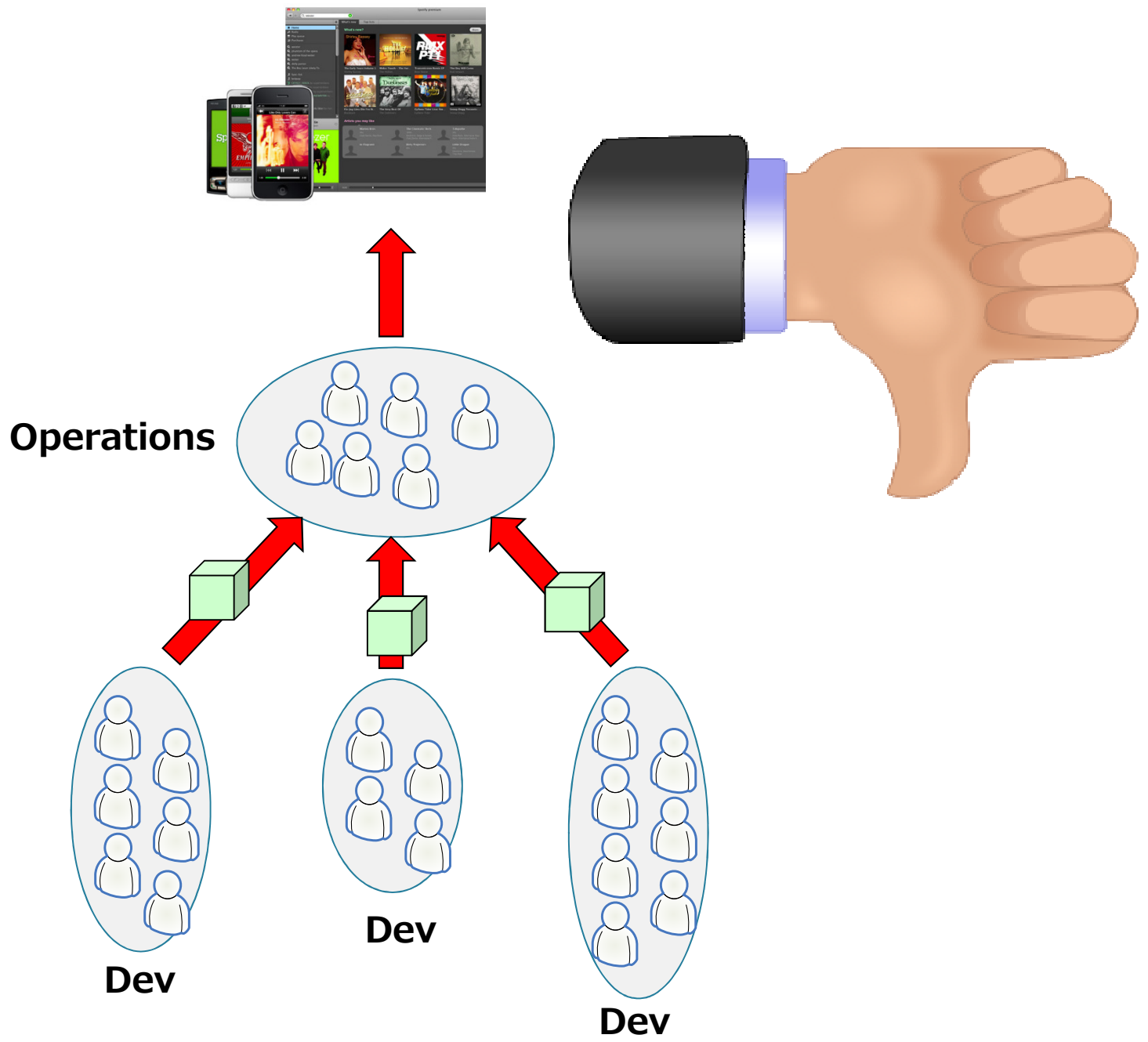
Squad

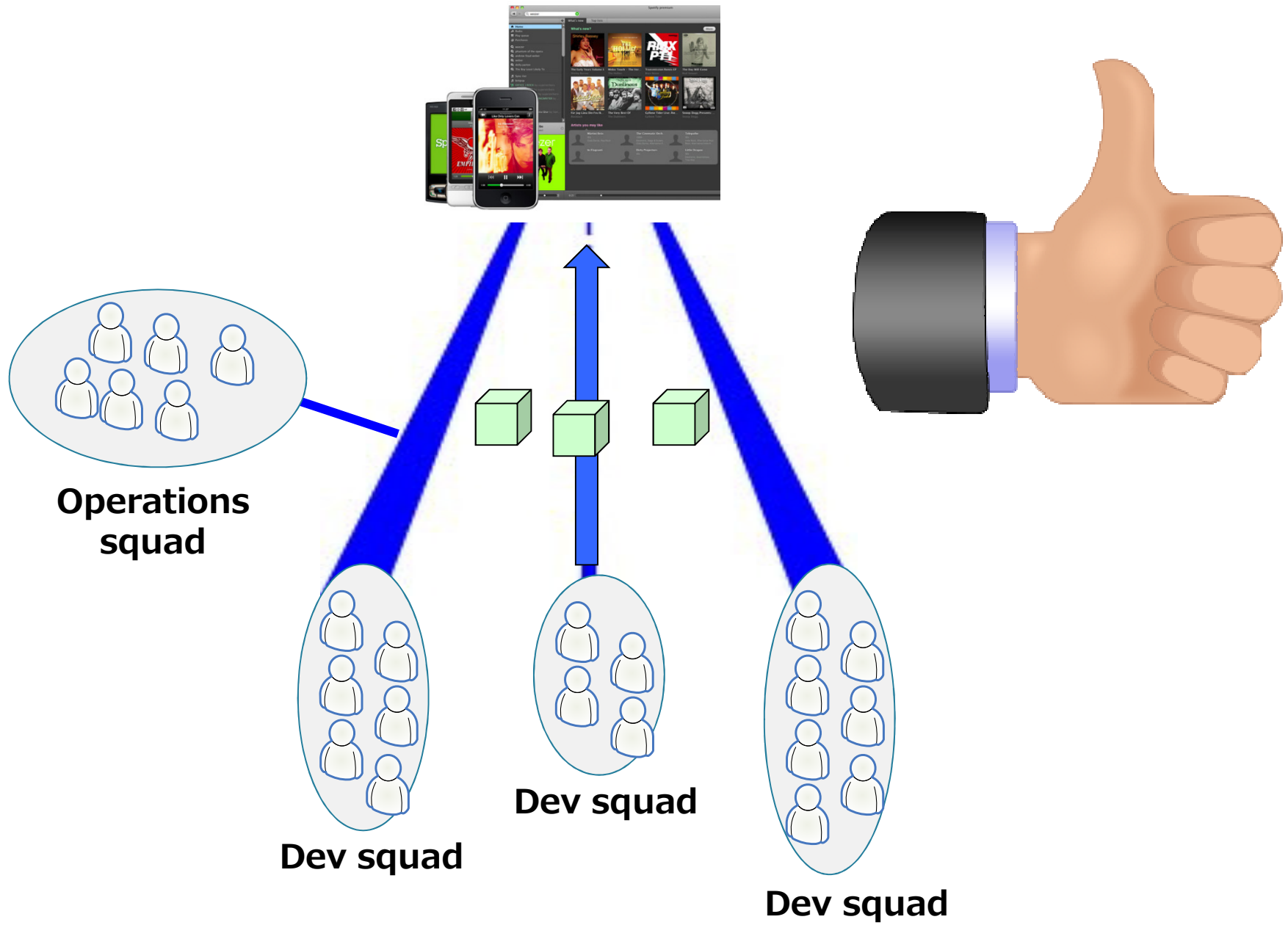


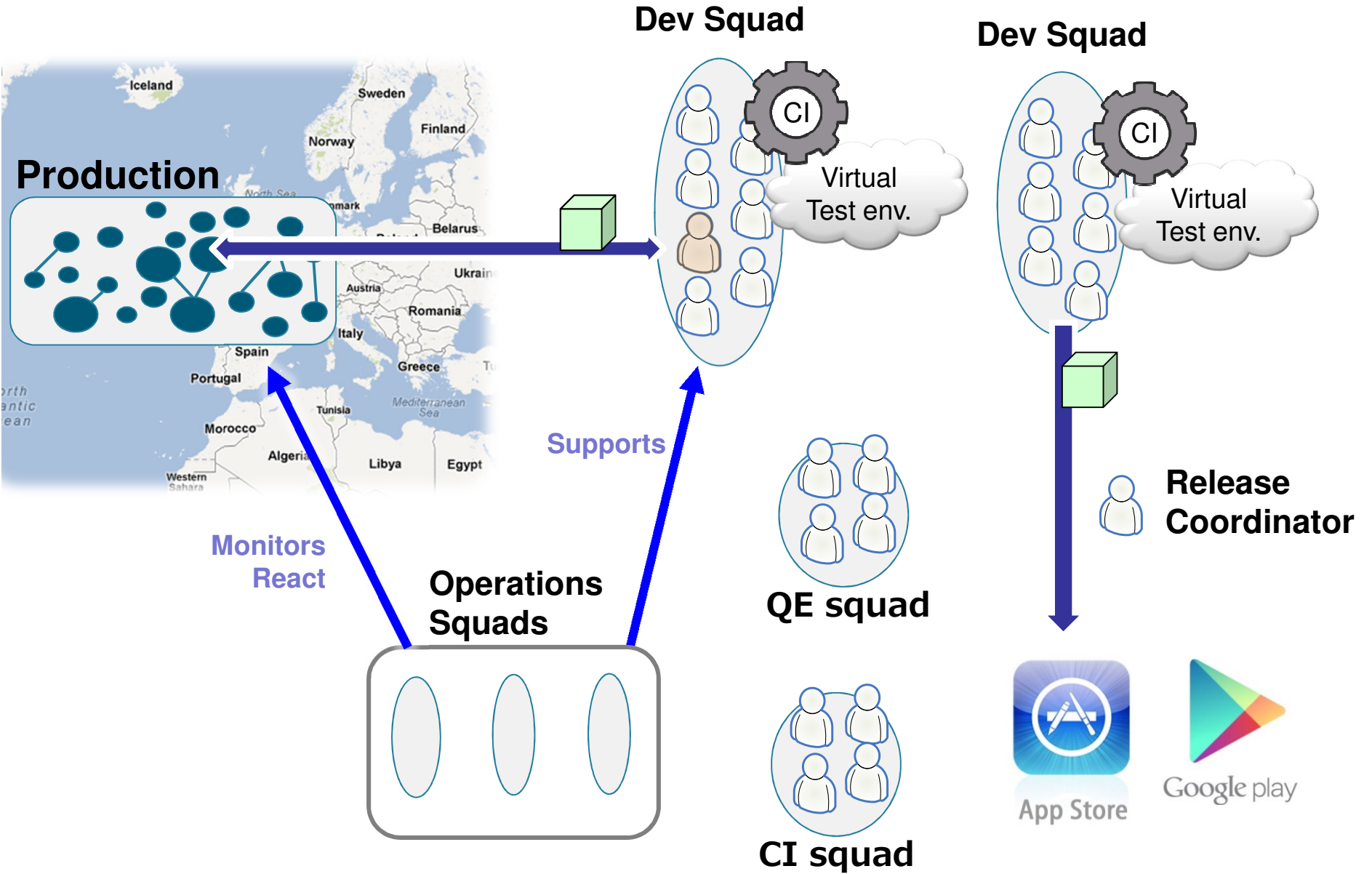
- ✓ **Autonomous with a mission**
- ✓ **Self-organizing, Cross-functional**
- ✓ **5-7 engineers, less than 10**
- ✓ **Stable**
- ✓ **Dedicated PO and Coach**
- ✓ **Owns process and quality**
- ✓ **Delivers to prod**
- ✓ **Operational responsibility and On-call**

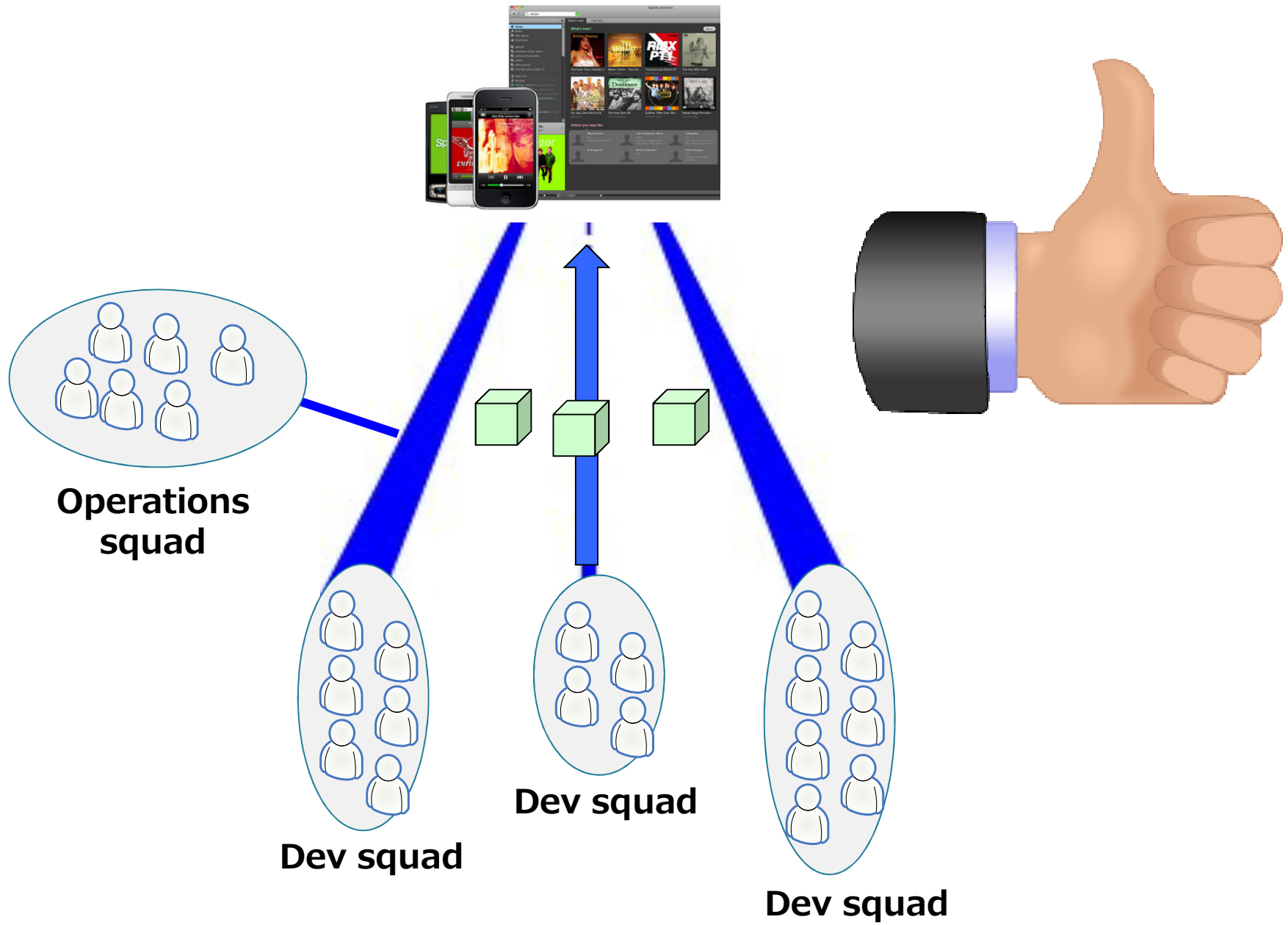
From Squad to Prod?







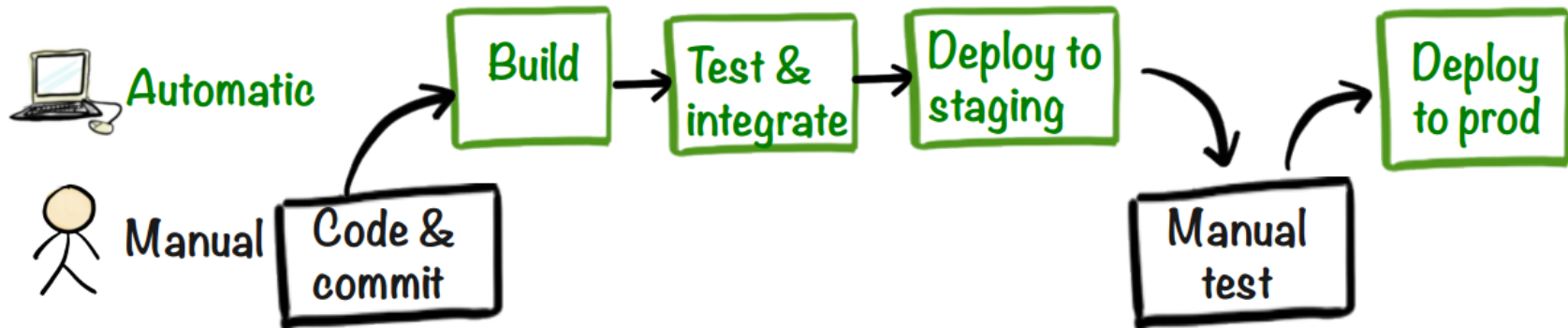




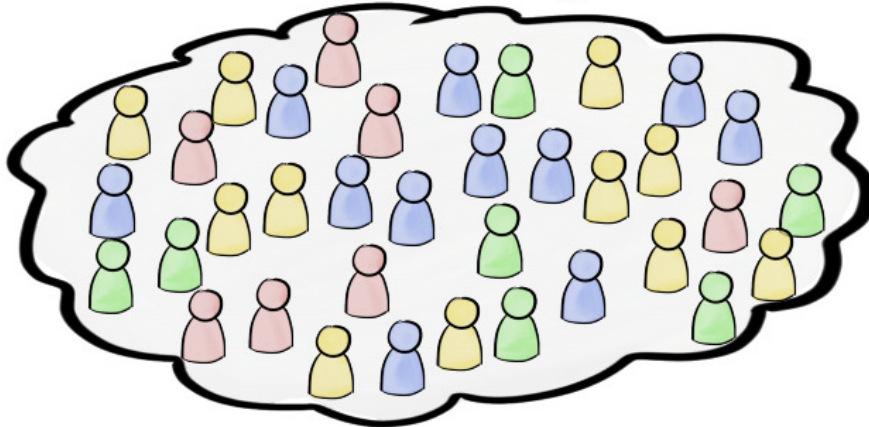
Big joint testing sessions



Pushing for Continuous Delivery



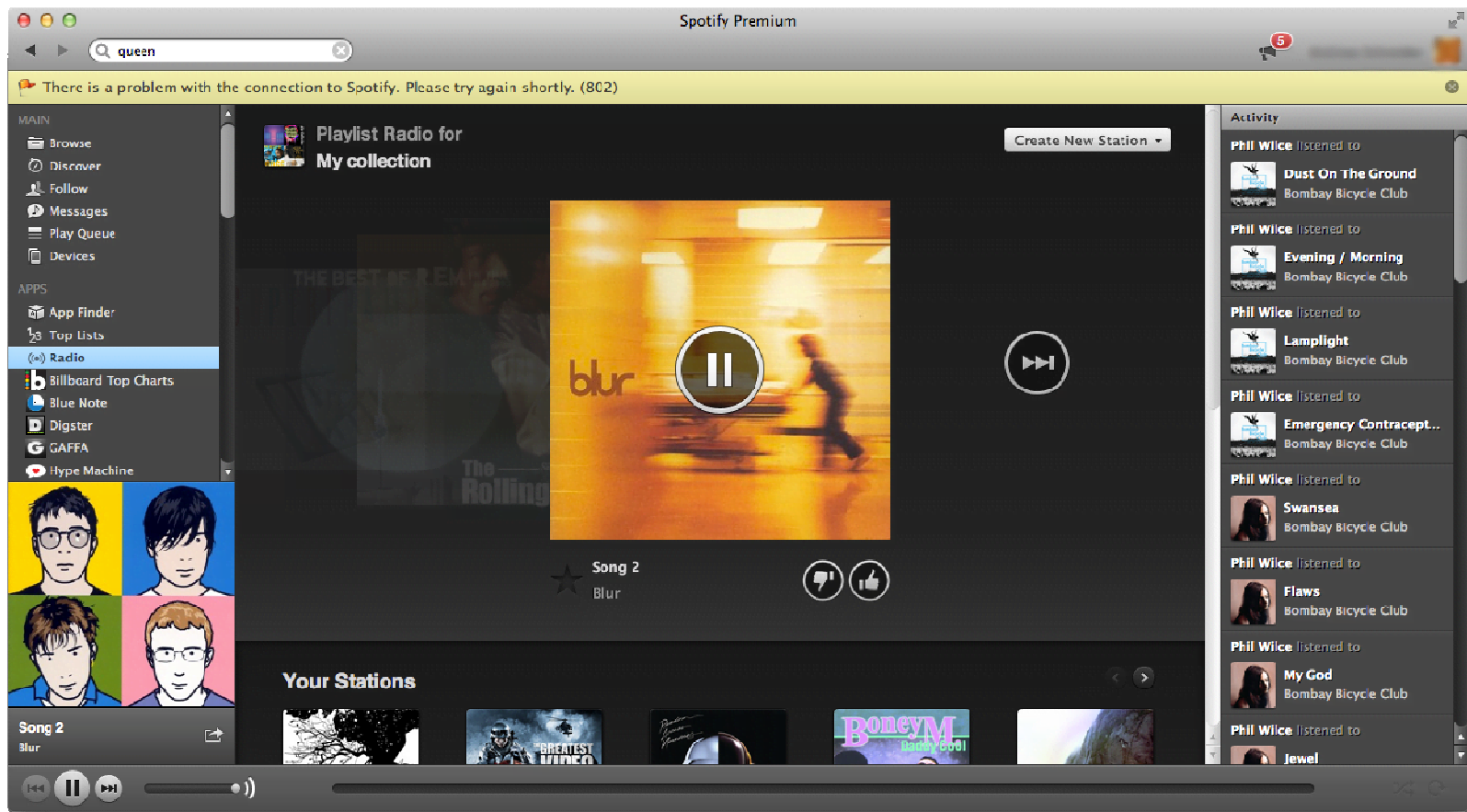
Continuous Delivery Guild



Commit-to-Deploy squad



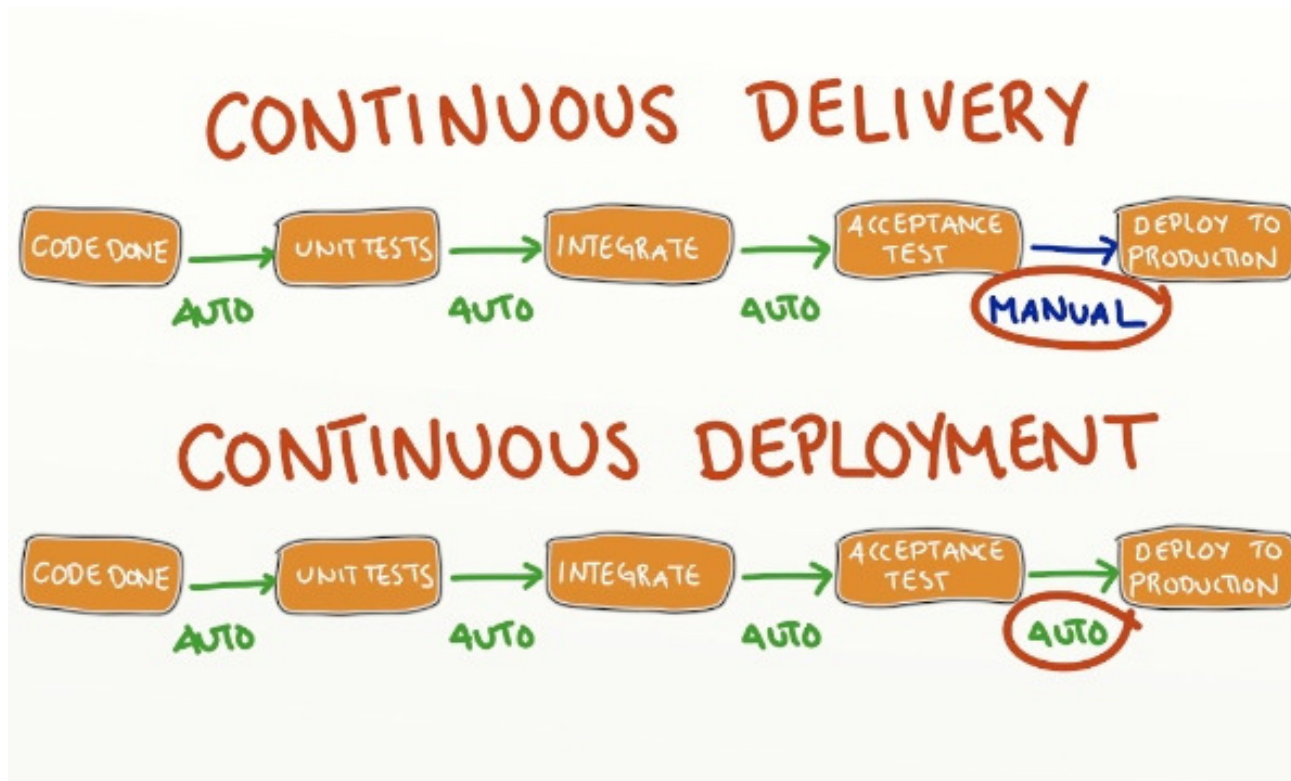
Monitoring, Operational Responsibility and On-Call



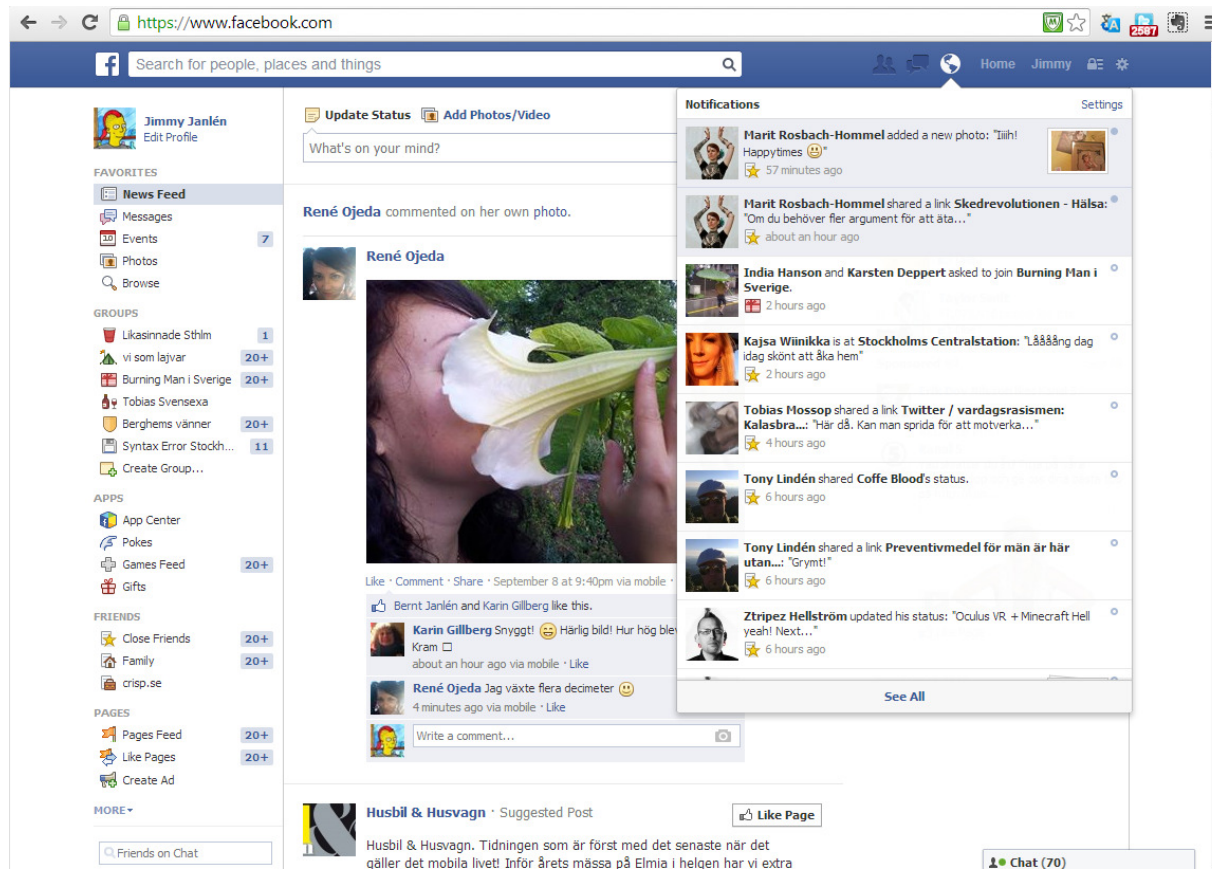
Current Trends & The Future

Release Pipeline

Continuous Delivery vs Continuous Deployment



Dark launching

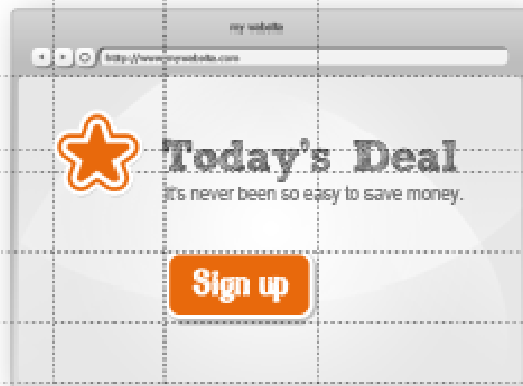


Automatic Rollback

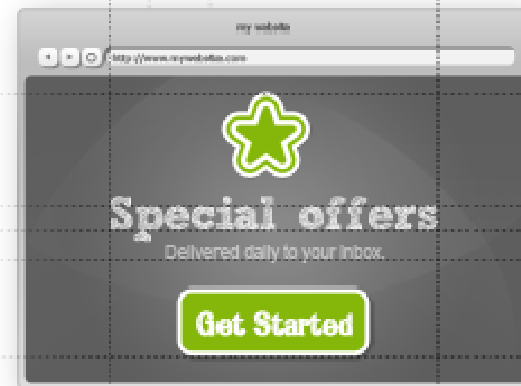


Increase
knowledge
runtime

A/B-testing



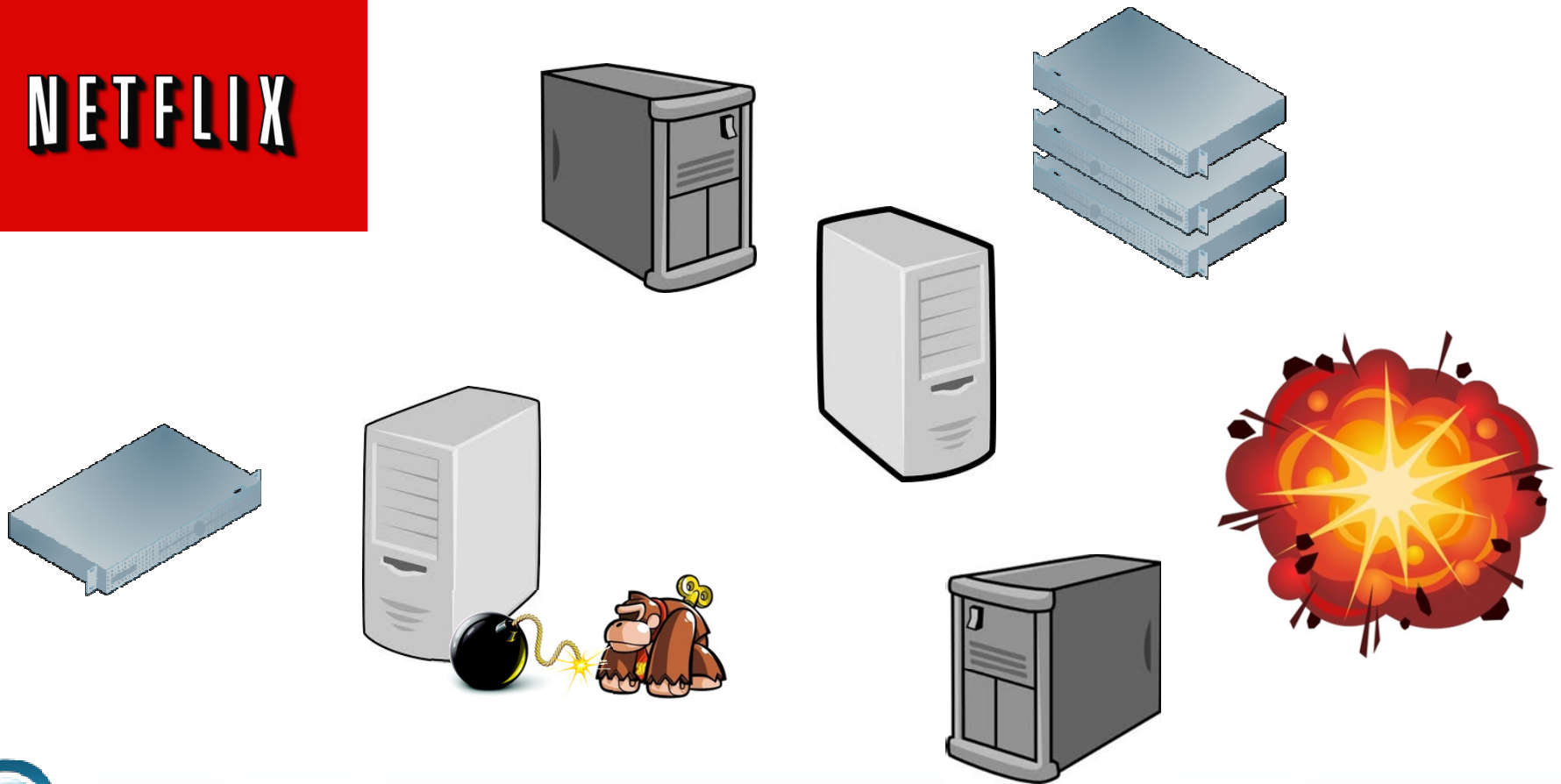
Page A



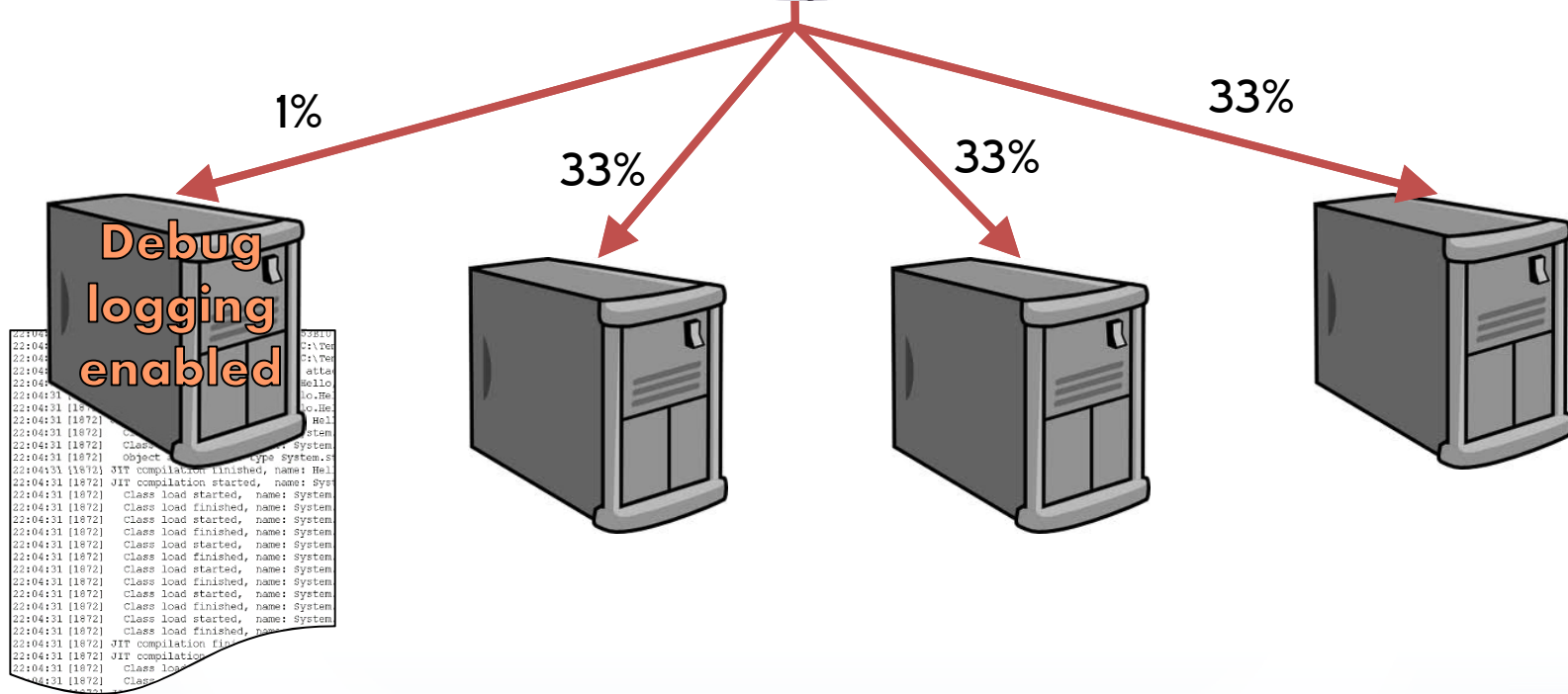
Page B

Destructive robot monkeys

NETFLIX



Massive logging in prod





Atlas Copco

~~Requirement spec.~~
Tests

Atlas Copco – TDD Example

Automated Unit Test

```
[Test]
public void Does_not_generate_drill_plan_if_view_is_invalid()
{
    ViewIsInvalid();
    _presenter.Generate();

    _drillPlanGenerator.AssertWasNotCalled(x => x.Generate(null), x => x.IgnoreArguments());
}

[Test]
public void Shows_error_message_when_trying_to_generate_from_invalid_input()
{
    ViewIsInvalid();
    _presenter.Generate();

    _dialog.AssertWasCalled(x => x.ShowValidationFailedMessage());
}
```

Atlas Copco – TDD Example

Automated System Description Documentation

Code + Grep + Sed →

Drill plan generation presenter:

- Previewing generates drill plan from the parameters in the view model
- Previewing adjusts the camera position after setting the new drill plan
- Shows error message when trying to generate preview from invalid input
- Does not show error message when trying to generate preview from valid input
- Does not generate drill plan and modify view for preview if view is invalid
- Selects first hole in generated drill plan when previewing
- Shows generation dialog when initialized
- Delegates drill plan generation to drill plan generator
- Generates drill plan from the parameters in the view model
- Saves generated drill plan
- Does not generate drill plan if view is invalid
- Shows error message when trying to generate from invalid input
- Opens drill plan view after successful generation
- Saves generated drill plan before opening drill plan view
- Does not close dialog when generating if view is invalid
- Closes dialog when generating if view is valid
- Updates view model with generated drill plan when previewing result



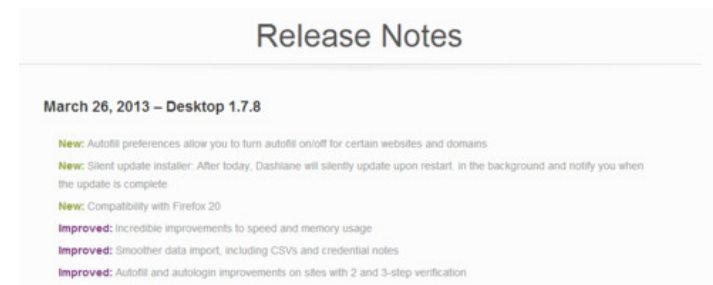
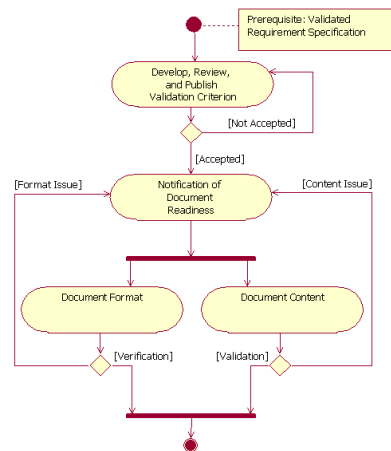
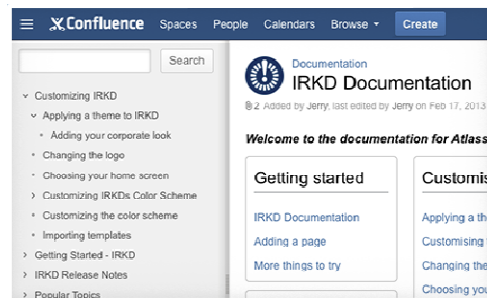
Atlas Copco – A-TDD Example

Automated Acceptance Test

```
[Category("UnderDevelopment")]  
[Category("AcceptanceTests")]  
[Test]  
public void Holes_can_be_removed()  
{  
    Given.  
        drill_plan_containing(plannedHoles: 4, drilledHoles: 0).  
        drill_plan_view_is_open().  
        two_holes_are_selected();  
  
    When.  
        user_presses_the_delete_holes_button();  
  
    Then.  
        number_of_visible_planned_holes_in_the_editor_is(2).  
        number_of_visible_planned_holes_in_the_visualization_view_is(2);  
}
```

Automatic documentation

- Automatic system descriptions and overview
- Automatic update of wikis
- Automatic release notes



Roles

More mature definition of testing skills and roles

Diversified expertise areas

- TA – Test Automation
- QA – Quality Assistance
- QE – Automation engineer
- Quality Coach / Mentor
- Release Coordinator/Conductor



Myth: Automation
replaces the tester

Myth: Automation will kill the tester?

BUSTED



- Automated tests are just fact checking!
- Skill still needed, broadens the team (XFT, end-to-end)
- Tester → QA/QE/TA/QC/etc.
- Strengthens the team's planning, design and ability to deliver with their input, expertise and experience
- Broadens the perspective (process, testability, etc.)

<http://agilamyter.wordpress.com>



Agila Myter BUSTED (och några bekräftade)

[Hem](#) [Om bloggen](#) [Om oss](#)

Om oss



Jagannath "Jagge" Tammeleht
Agil testare, Scrum Master, testledare. föreläsare och thought leader på **Claremont AB** (konsult). Co-driver av **testzonen**.



Jimmy Janlén
Agile/Lean coach och lärare på **Crisp** (konsult). **Bloggar**, håller seminarier och är intresserad av **grafisk dokumentation** och continuous discovery.



Alexander Tarnowski
Utvecklare, arkitekt och byggmiljökonstnär på **Crisp** (konsult). Skriver på en bok, **"Developer Testing"**.



Therése Ressel
Agil testare, testledare och Scrum Master på Kambi Sports Solutions. Skribent för **testzonen**.

"Myter är som kamouflerad kollektiv visdom, det finns alltid något att lära. Ingen rök utan eld."



SÖK

type and press enter

SENASTE MYTER

- > Myt #3: Agila team behöver inga testare
- > Myt #2: Regressionstest hinns inte med
- > Myt #1: Alla agila testare måste vara tekniska

ÄLDRE MYTER OM...

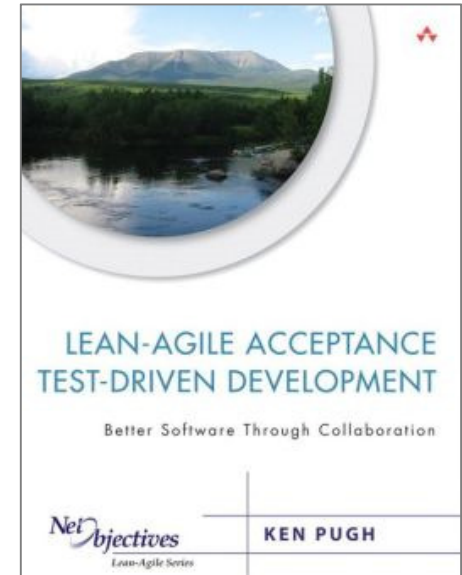
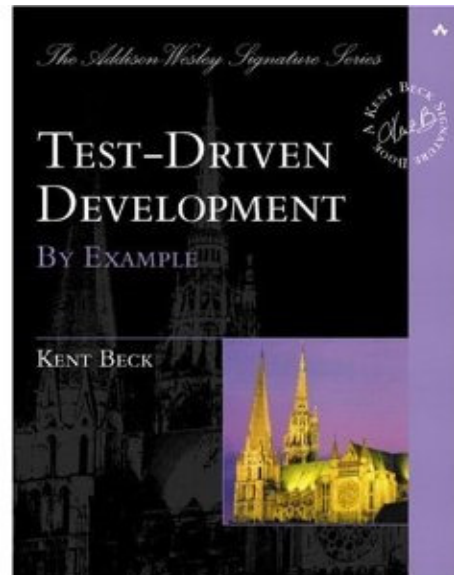
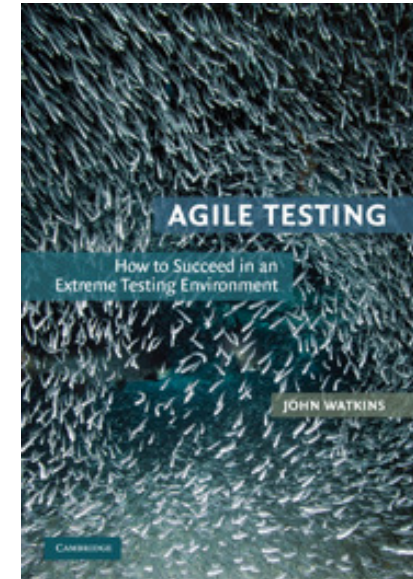
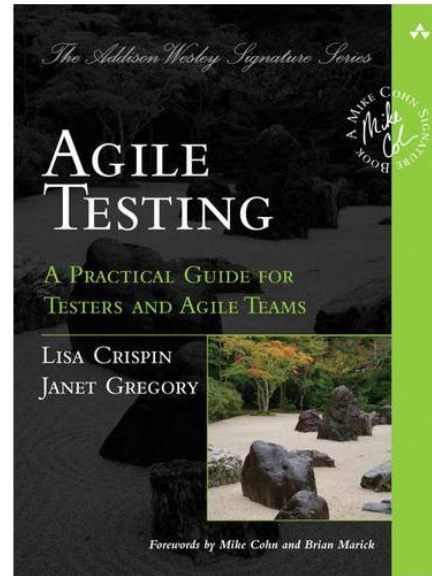
- > Agil Testning

Thank you!





Reading...



Me @ Online



Crisp
<http://www.crisp.se/konsulter/jimmy-janlen>



Twitter: @JimmyJanlen



LinkedIn: JimmyJanlen
<http://se.linkedin.com/in/jimmyjanlen>



jimmy.janlen@gmail.com



YouTube: Crisp Agile Academy
<http://www.youtube.com/crispagileacademy>



Swedish Blog
"Den Scrummande Konsulten"
<http://jimmyjanlen.wordpress.com>



Jimmy Janlén



Crisp's Blog
<http://blog.crisp.se>



HOME ABOUT

Får du allt av Agile?
Crisps coacher hjälper dig att slå in på vägen av ständig förbättring.



Crisp's Blog

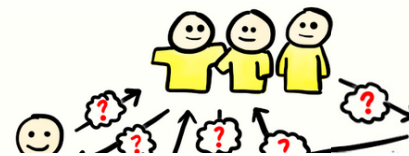
from the Crisp Consultants

Role Expectation Mapping

Posted on March 11, 2014 – 1:44 am by Jimmy Janlén

Role Expectation Mapping is a series of workshop that explores, clarifies and establishes which expectations members of a group, team or project have on each other.

If you suspect that collaboration is undermined because of mismatch of expectations between people, then this exercise could boost the team's ability to collaborate efficiently together. It is also a powerful way to jump start a new team and give them a structure to relate to.

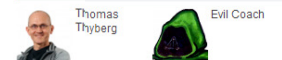


[Subscribe via RSS](#)

Guest Contributors



Blog Authors



Jimmy

Janlén

119

Me @ Online



Crisp
<http://www.crisp.se/konsulter/jimmy-janlen>



Twitter: JimmyJanlen
<http://twitter.com/#!/jimmyjanlen>



LinkedIn: JimmyJanlen
<http://se.linkedin.com/in/jimmyjanlen>



Email:
jimmy.janlen@gmail.com



Swedish Blog
"Den Scrummande Konsulten"
<http://jimmyjanlen.wordpress.com>



Crisp's Blog
<http://blog.crisp.se>



HOME ABOUT

Får du allt av Agile?

Crisps coacher hjälper dig att slå in på vägen av ständig förbättring.



Se alla coacher >

Crisp's Blog

from the Crisp Consultants

Role Expectation Mapping

Posted on March 11, 2014 – 1:44 am by Jimmy Janlén

Role Expectation Mapping is a series of workshop that explores, clarifies and establishes which expectations members of a group, team or project have on each other.

If you suspect that collaboration is undermined because of mismatch of expectations between people, then this exercise could boost the team's ability to collaborate efficiently together. It is also a powerful way to jump start a new team and give them a structure to relate to.



Search

Subscribe via RSS

Guest Contributors



Christopher Avery



Ellen Gottesdiener



Jeff Gothelf

Blog Authors



Thomas Thyberg



Evil Coach



Jimmy Janlén