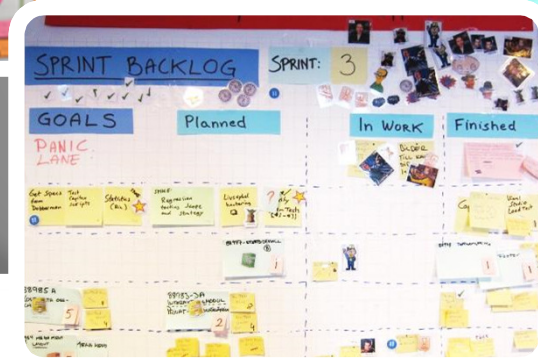
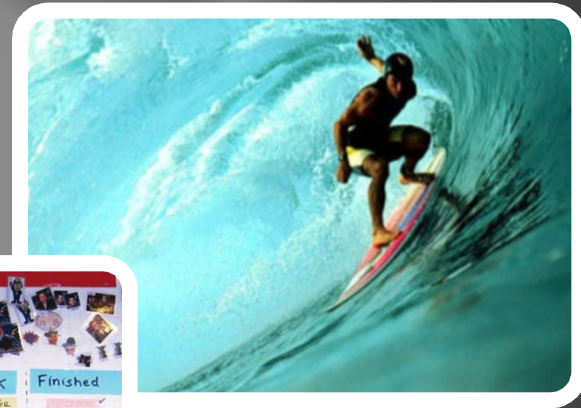
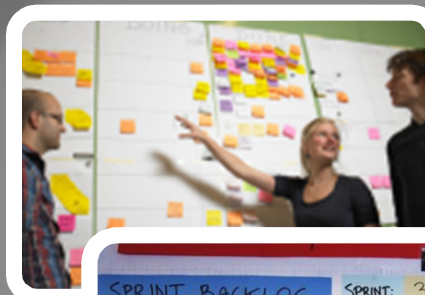


# Agile Testing

## Will automation kill 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](http://www.crisp.se/konsulter/jimmy-janlen)



### History...

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

### Work as...

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

### Experience...



Jimmy Janlén

# 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](mailto:jimmy.janlen@gmail.com)



**YouTube: JimmyJanlen**  
<http://www.youtube.com/user/JimmyJanlen>



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



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

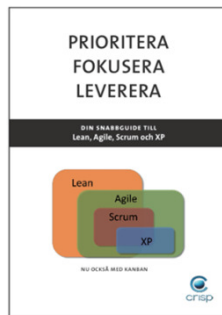
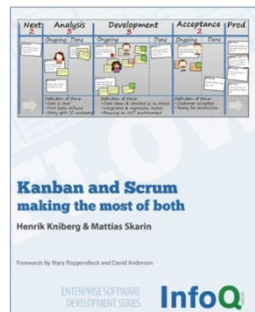
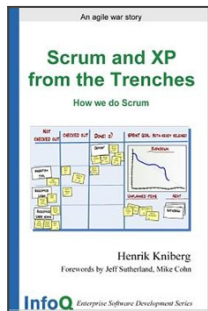




Crisp is an employee owned company known for agile courses with internationally renowned teachers and experienced agile developers and coaches.



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!



# Context

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





# Being Agile & Lean

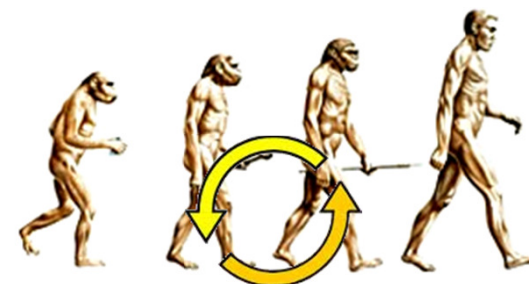
# Agile is...

## Working & Valuable Products.

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

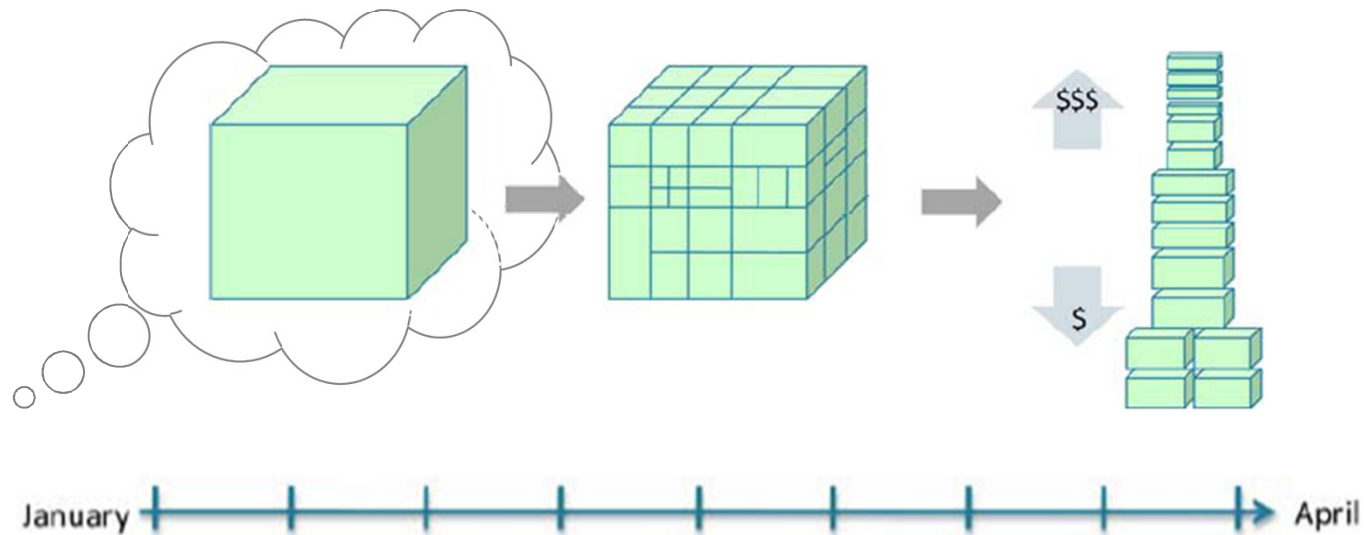


Welcoming change.  
Managing change.  
Respond to change.



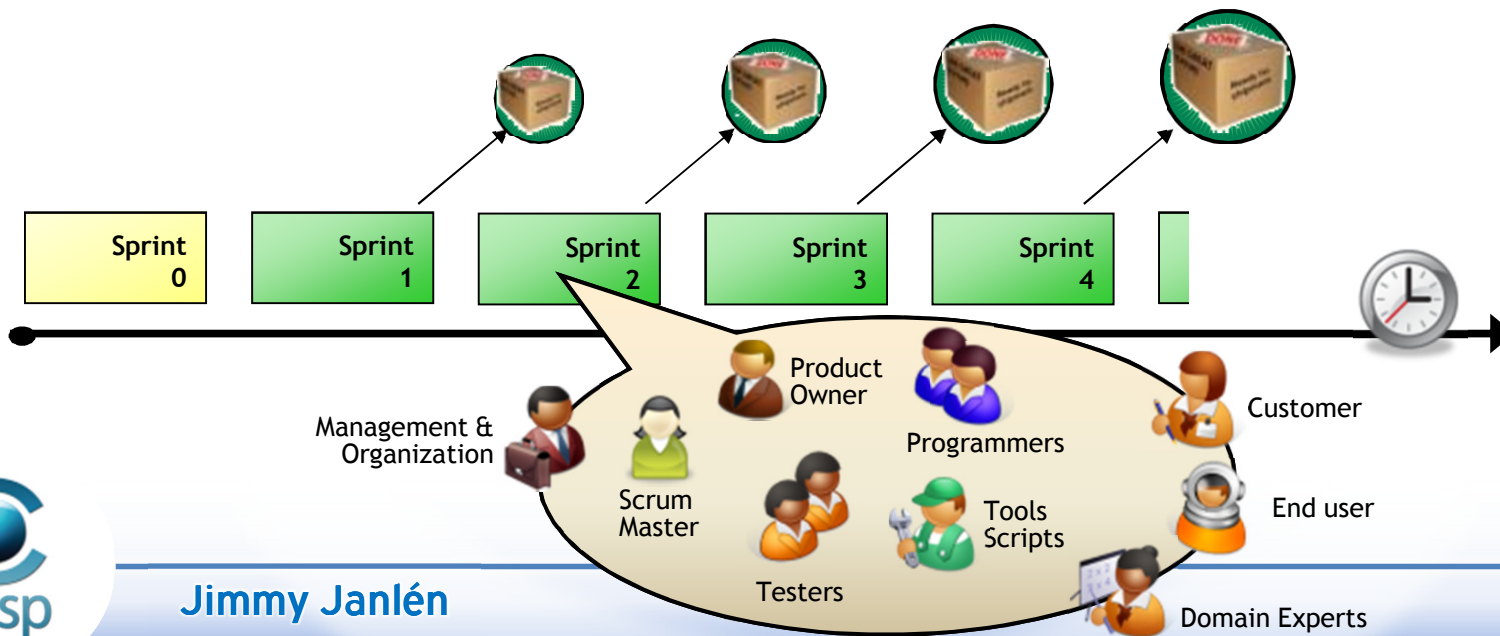
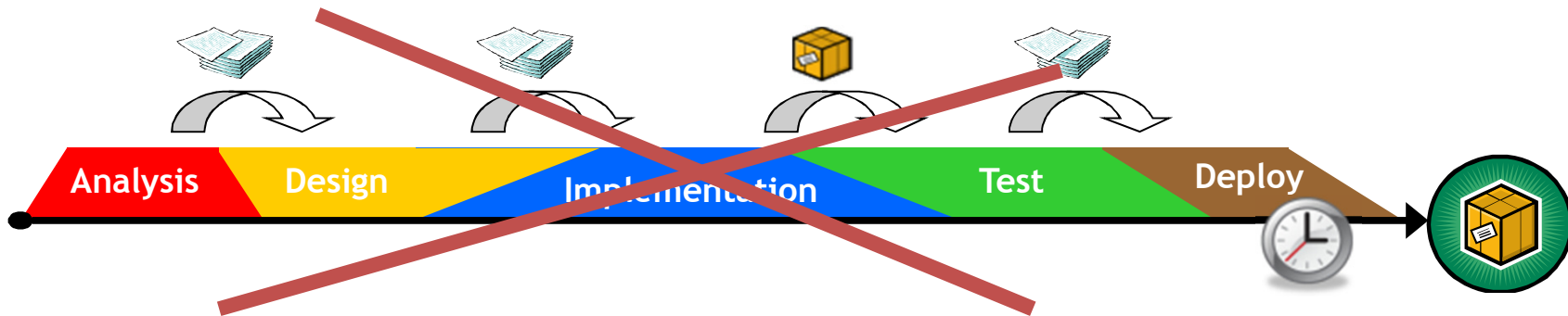


# Iterative incremental development



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

# Incremental delivery and short iterations

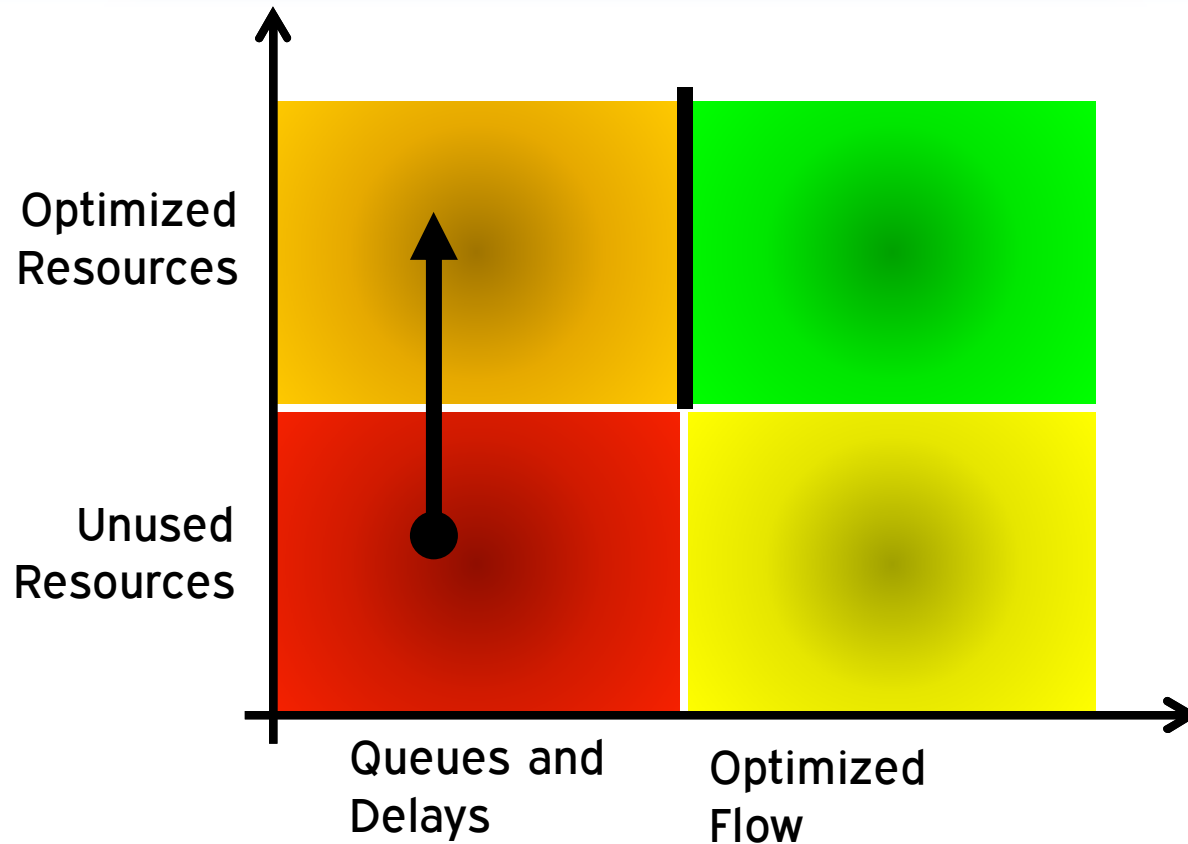


# Lean - Summarized and Simplified

- Maximize customer value
- Minimize delivery time
- Minimize waste

**Continuously  
improve (kaizen)**

# Lean - Summarized and Simplified



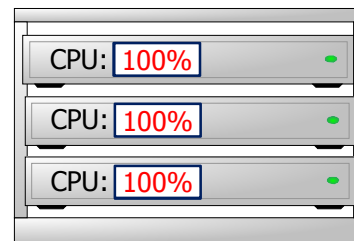
# Lean - Summarized and Simplified



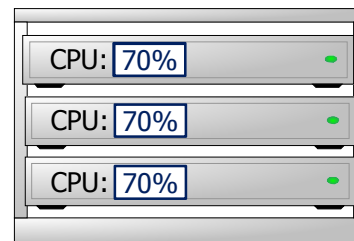
Road utilization = 100%  
Throughput: Terrible



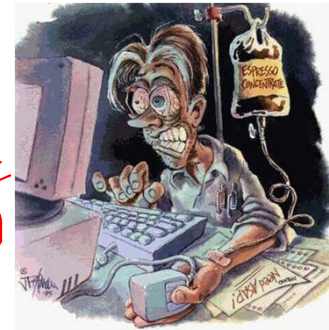
Road utilization  $\approx$  70%  
Throughput: Fast



Server utilization = 100%  
Response: Slow



Server utilization = 70%  
Response: Fast

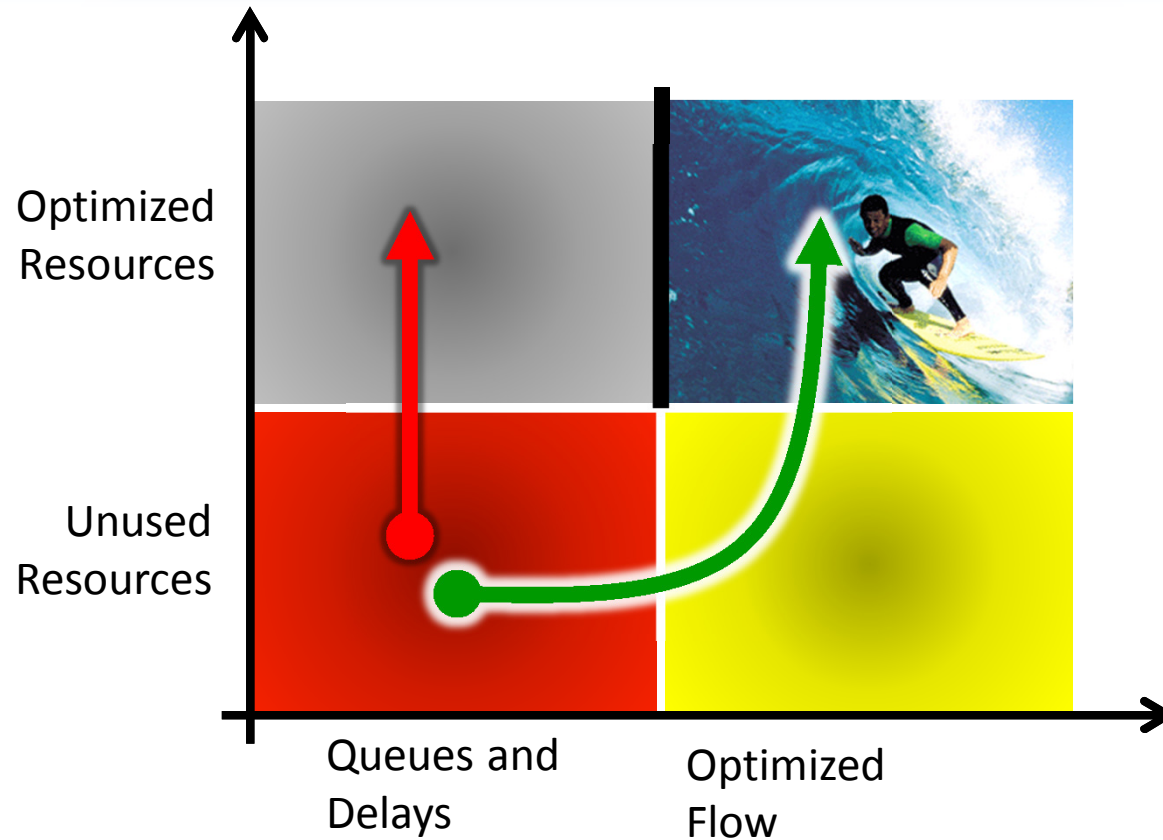


Staff utilization = 100%  
Delivery: Slow



Staff utilization  $\approx$  70%  
Delivery: Fast

# Lean - Summarized and Simplified

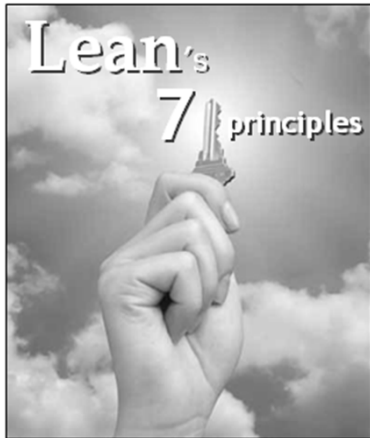


# Lean Software Development

## Lean's 7 principles

1. Eliminate Waste
2. Build Quality In
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimize the Whole





1. Eliminate Waste
2. Build Quality In
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimize the Whole

## The seven wastes of software development

### Waste: Partially Done Work

Examples:

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

### Waste: Extra features

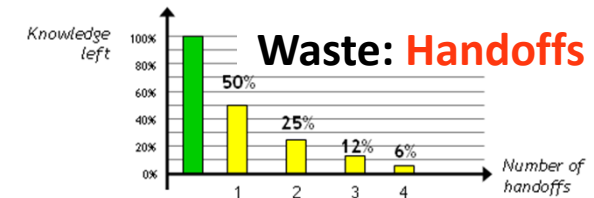
*Don't predict nor guess!*



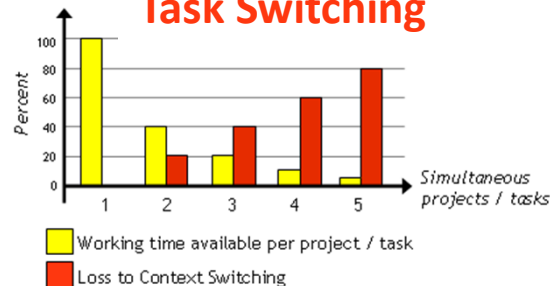
### Waste: Relearning

Examples:

- Poor knowledge sharing
- Poor documentation
- Repeating mistakes



### Waste: Task Switching



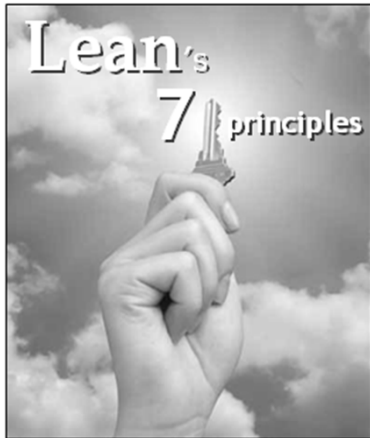
### Waste: Delays and Motions



### Waste: Defects







1. Eliminate Waste
2. Build Quality In
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimize the Whole

## The seven wastes of software development

### Waste: Partially Done Work

Examples:

- Uncode
- Unsync
- Untest
- Undoc
- Undep

**Waste:**  
**Wishful Thinking**



### Waste: Extra features

**Waste: Fear**



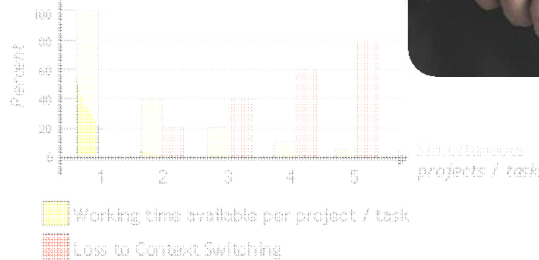
### Waste:

Examples:

- Poor
- Poor documentation
- Repeating mistakes



### Waste:



### Delays

### Waste:



# Agile Testing

# Why the confusion?

# Agile Manifesto

[www.agilemanifesto.org](http://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

- ★ Prevent defects
- ★ Provide fast feedback
- ★ Prevent partially done work  
(minimize batches and multitasking)
- ★ Build Quality *into* the product and process
- ★ Enable courage in the team
- ★ Build confidence to release fast and often
- ★ Remove waste  
(Handoffs, Defects, Wishful Thinking, Relearning)

# An Agile Test Approach...

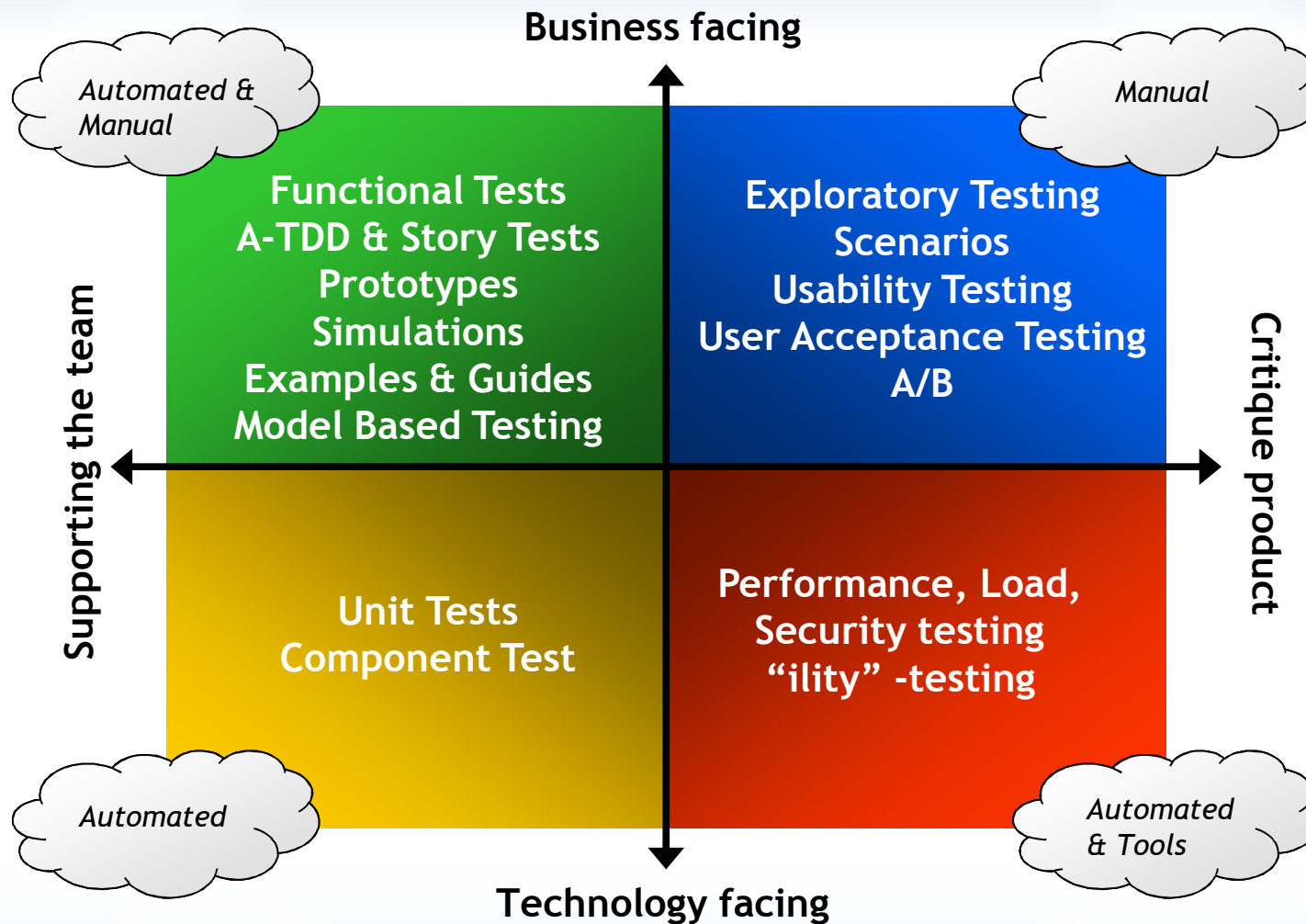


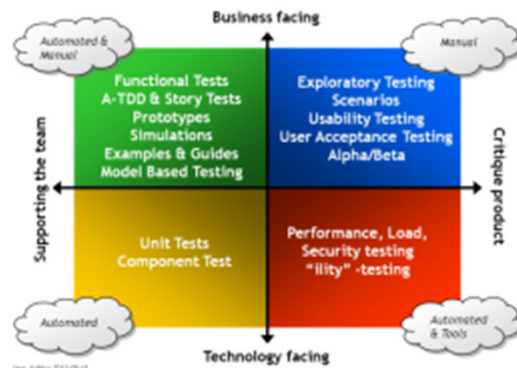
## Adheres and obeys to the principles of Agile & Lean...

- ☑ Embraces changes late in development and manage uncertainty in upcoming features
- ☑ Handles and utilizes that testing and development happens at the same time in close collaboration
- ☑ Fits within a sprint and works towards potential release (that is tested and done)
- ☑ Uses lightweight test techniques
- ☑ Incorporates and utilizes a cross-functional team
- ☑ Works with a test base that evolves with knowledge
- ☑ Introduces demands on test environment and support processes
- ☑ Is continuously questioned and improved



# Agile Test Quadrants





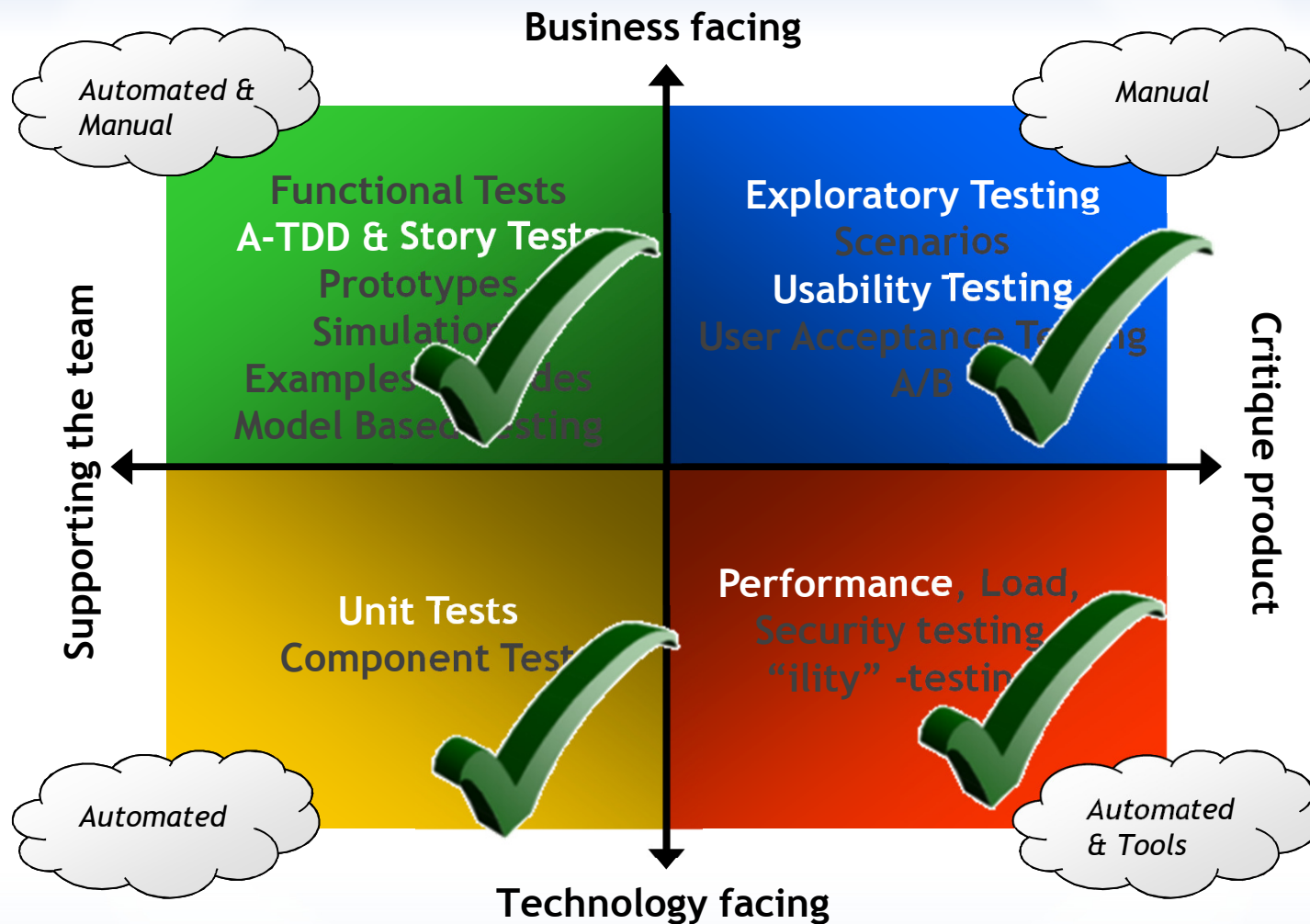
How can we do  
**ALL** of this – **EACH** Sprint?

# Decrease scope



**KISS – Keep It Simple Stupid**

# Agile Test Quadrants



**Being DONE  
VS  
Achieving DONE**

# Being DONE vs Achieving DONE



## Definition Of DONE

**DONE**

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

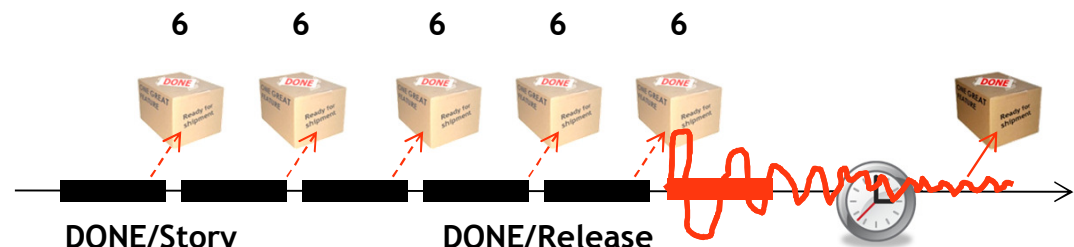


DONE/Story

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

DONE/Release

✓ Already fulfilled



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 DONE for **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!



# The scope of test is increased



*The multi-tool, multi-purpose,  
multitasking power tester?*

# If we were out adventuring...



# 10 principles for the Agile tester

- Provide continuous feedback
- Deliver value to the customer
- Enable face-to-face communication
- Have courage
- Keep it simple
- Practice continuous improvement
- Respond to change
- Self-organize
- Focus on people
- *Enjoy!*



Lisa Crispin



Janet Gregory

# Myths on Agile & Testing

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

## BUSTED

IT'S EASIER!

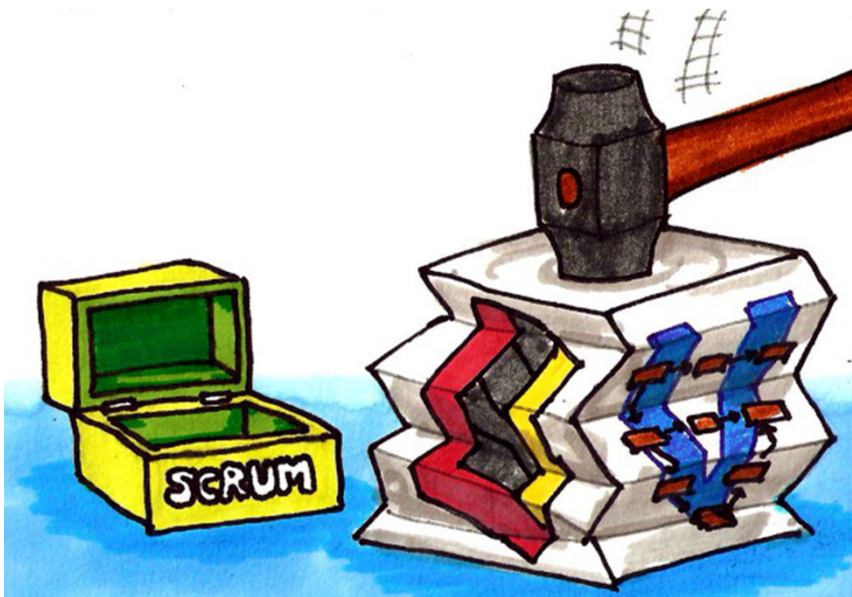
- Scope is smaller = Less to estimate
- Smaller deliverables, less risk
- Testers part of planning and design discussions
- Short learning cycles, estimations get better every sprint



# 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

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

# 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)

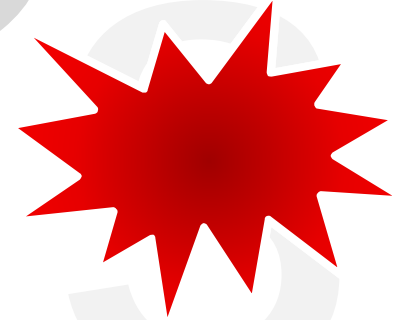
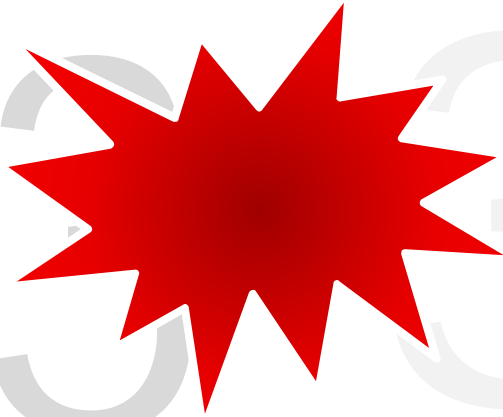


Delays opened up for Scoop creep...



# ZERO BUG POLICY





**RELEASE EVERY SPRINT!**



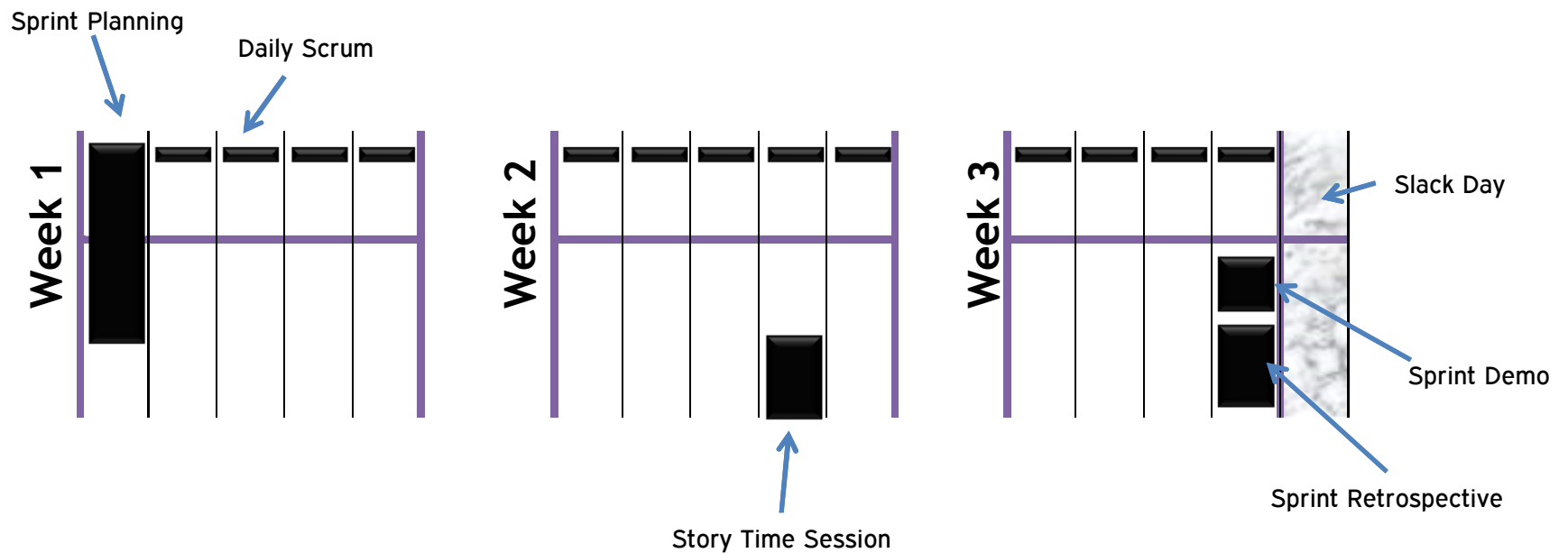


# 3 XFT TEAMS

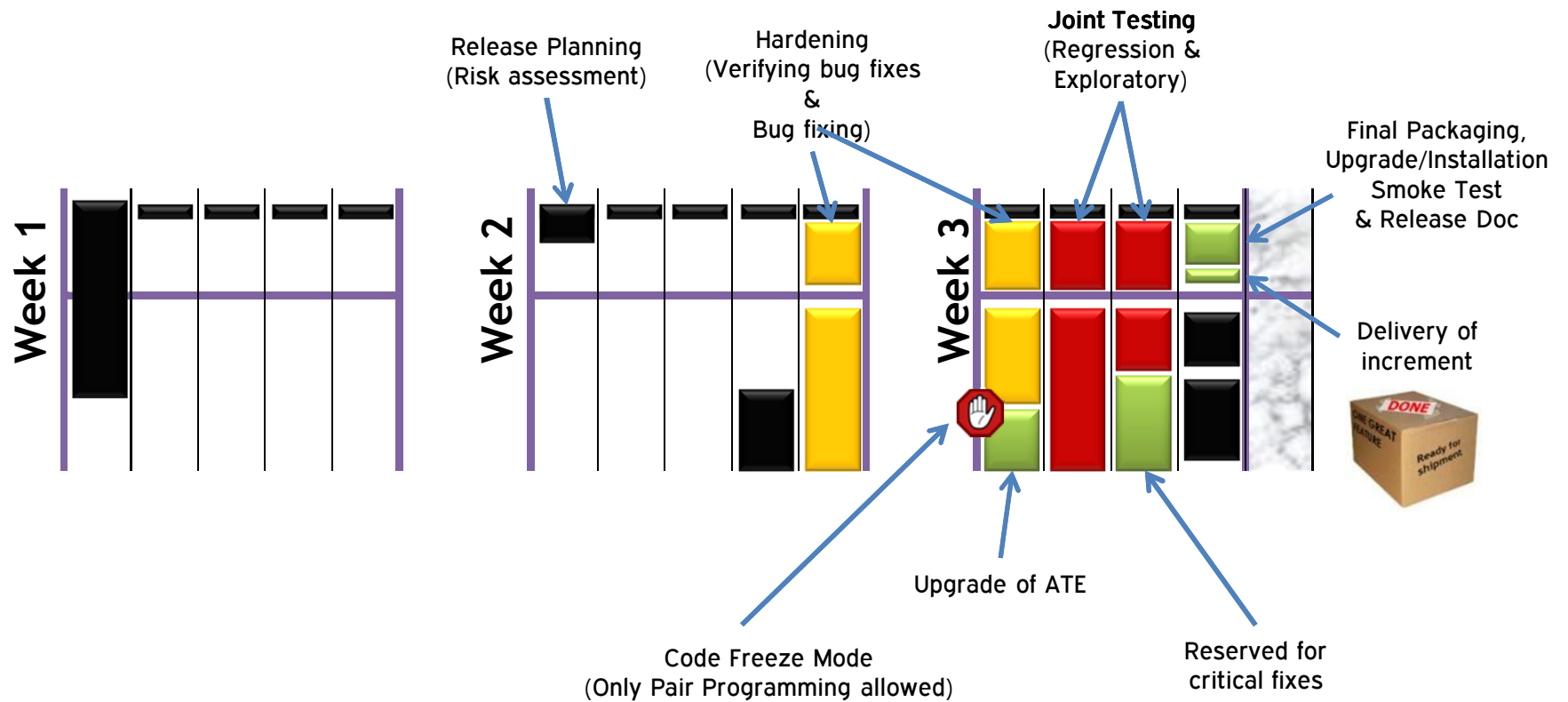




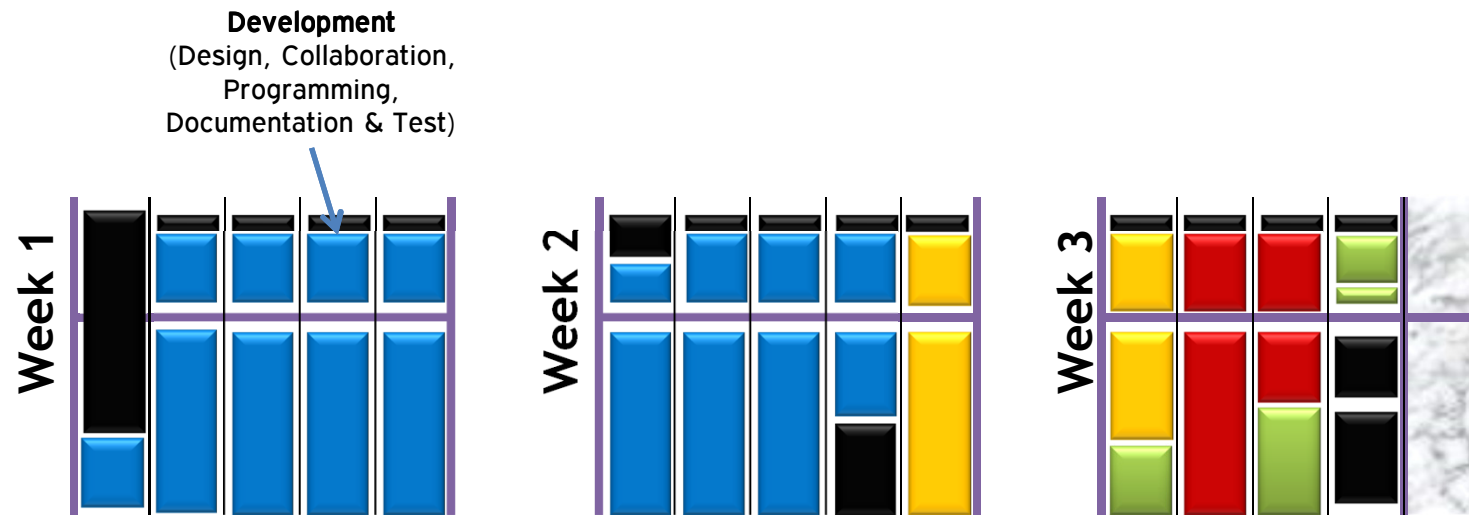
# Sprint Test & Release Strategy Example



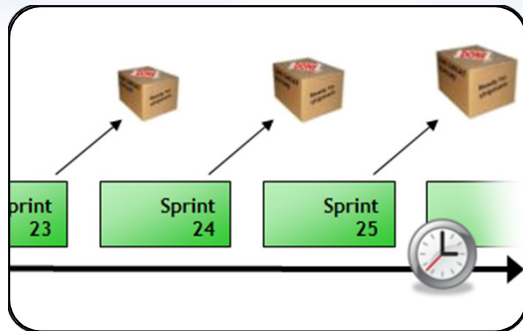
# Sprint Test & Release Strategy Example



# 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”

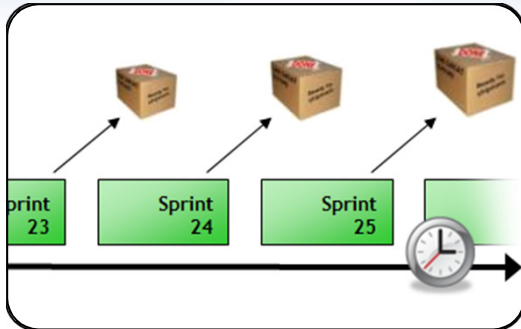


Less Defect-legacy  
(avg. age 30 days)



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 3<sup>rd</sup> week)**

# SIEMENS

medical

Large Scale Scrum &  
Separate Test Dep.

# Siemens Healthcare

# SIEMENS



## Soarian®



Soarian®: 1000 people, 3 major versions

Largest Scrum project:  
500+ people, 35 development teams, 5 countries

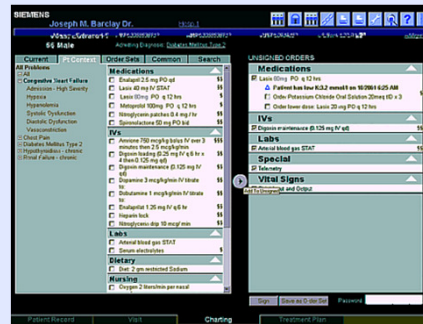
Many customers  
 Many wills  
 One application



Many customer



Different markets



Soarian®



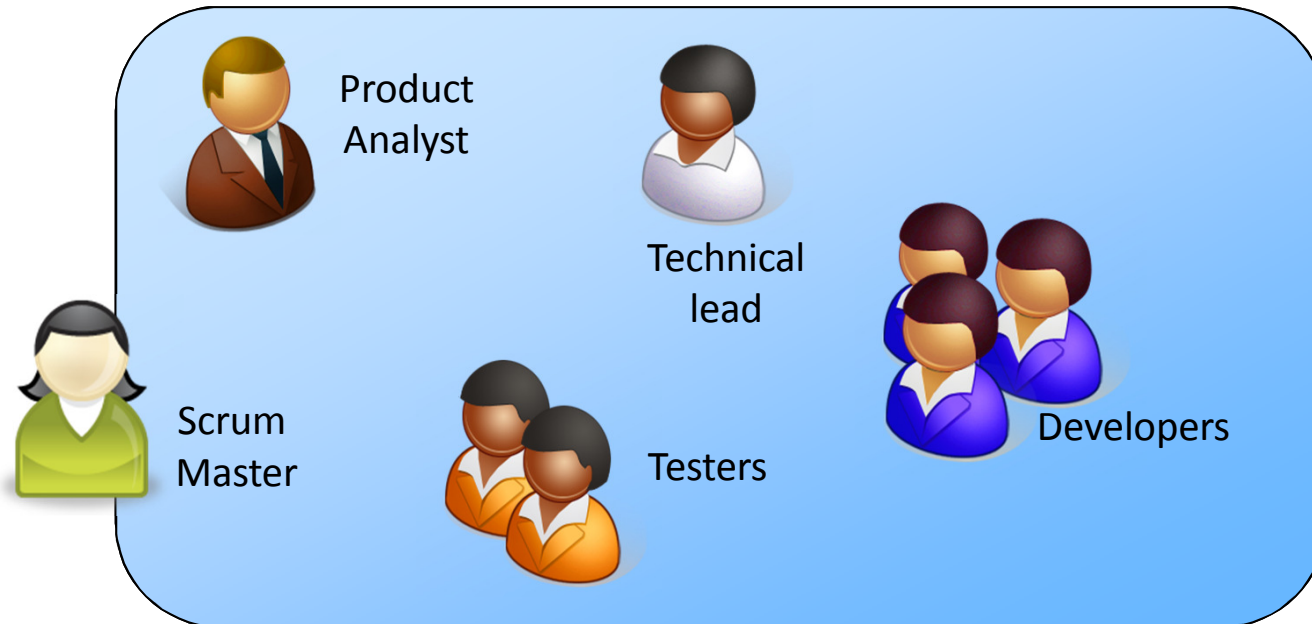
Complex needs



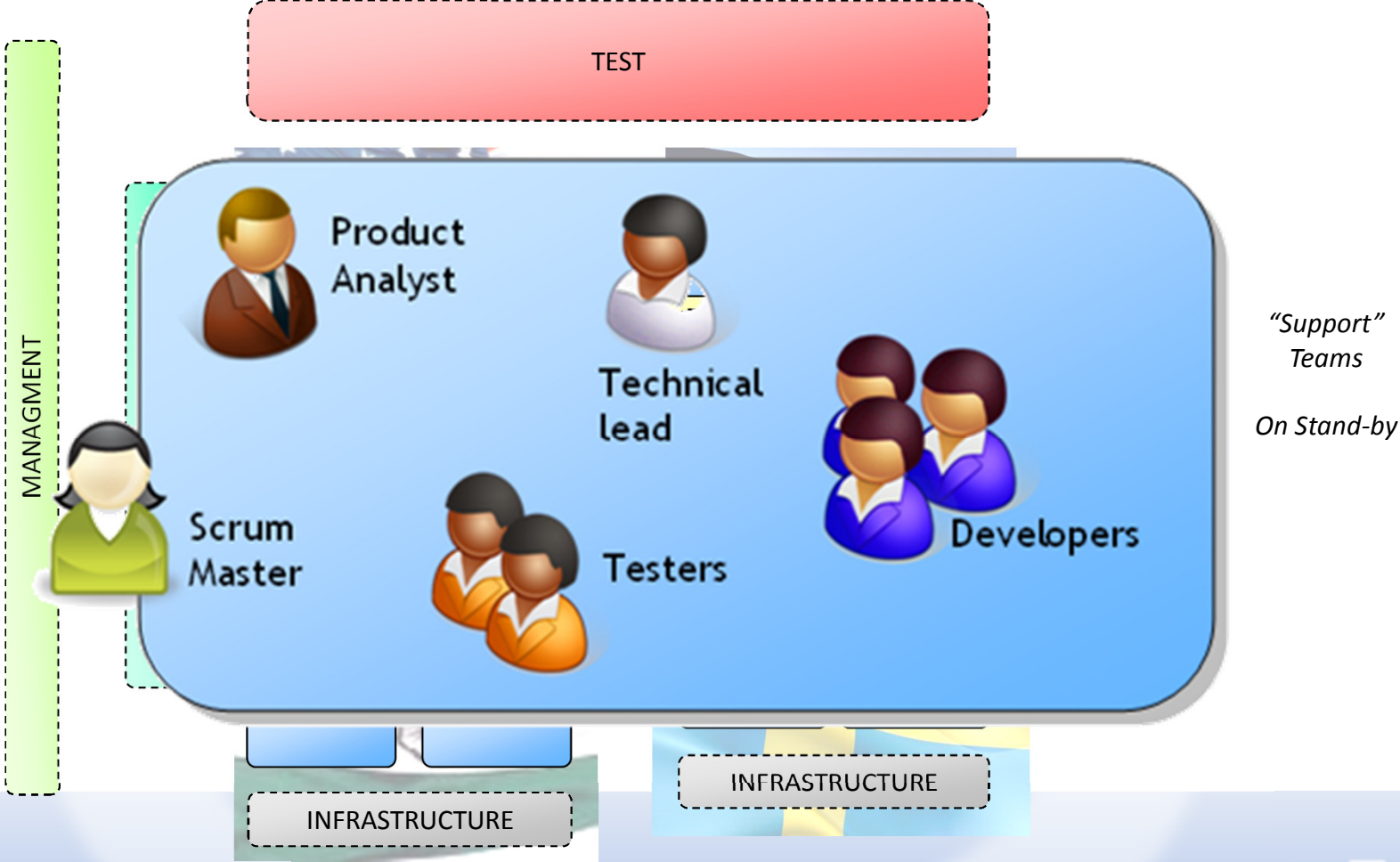
Changing legal demands



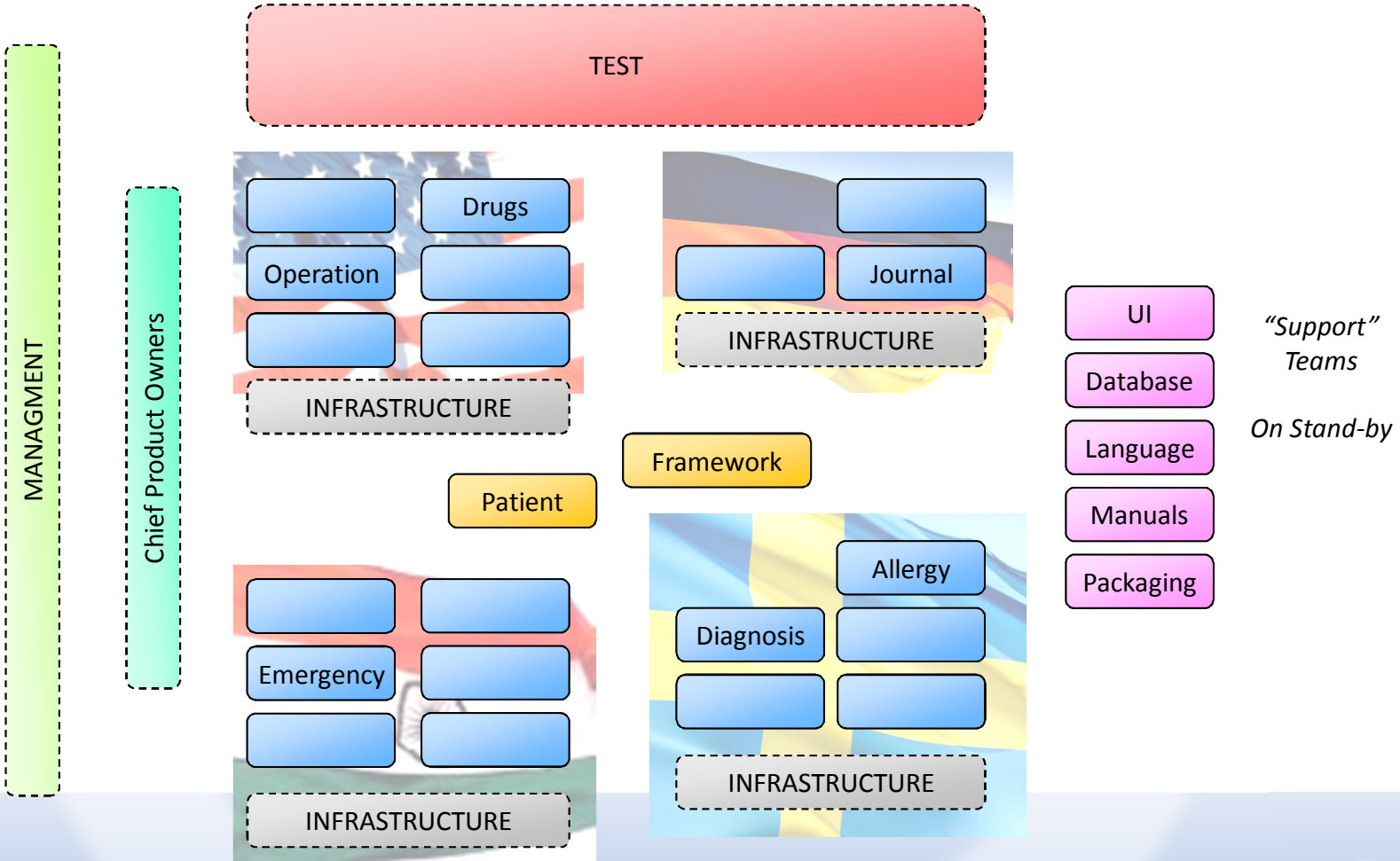
# A Typical Team



# Team focused organization

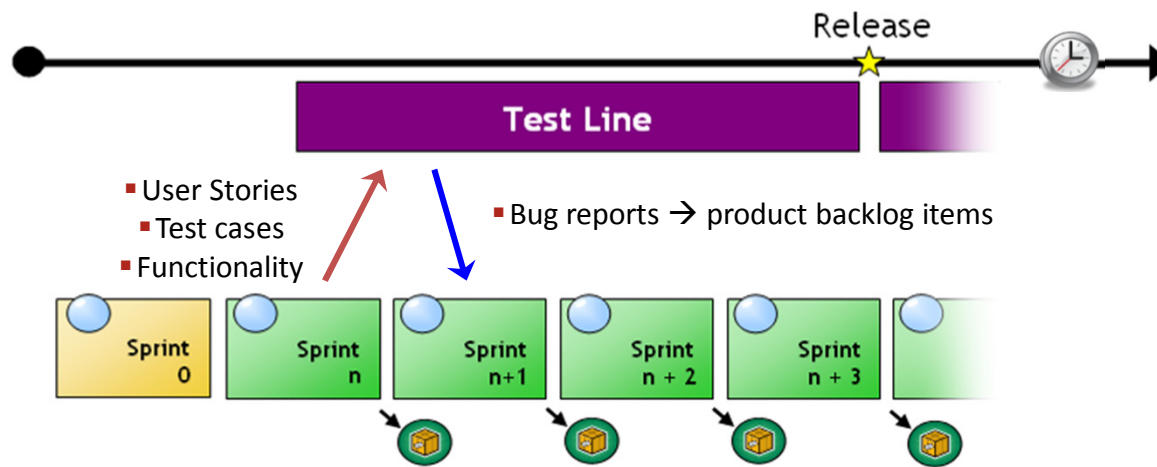


# Team focused organization





# Independent QA Team?



- Independent check and audit
- Unbiased and outside view
- Works toward a release date
- Economical restraints can force out-of-Sprint testing



Might create a sense of “safety-net” within the development team



Tacit knowledge is lost. Hand-offs and Relearning.



Heavier load on documentation.



Different goals → “We-them” mentality



Consider carefully what to test “outside”.



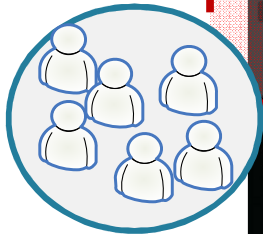
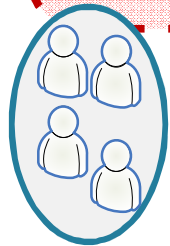
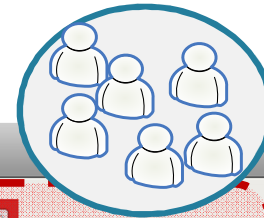
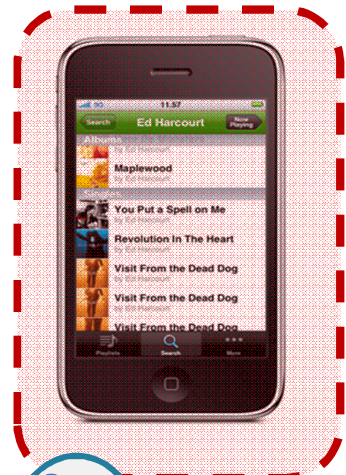
Spotify®

End-to-end  
responsibility

# Autonomy



# Example squads

A screenshot of the Spotify Premium desktop interface. The interface is framed by a red dashed border. Several blue circles highlight specific areas: one at the top right, one on the left side, one at the bottom left, and one at the bottom center. The interface shows a search bar, a sidebar with navigation options, a main content area with album covers (including "Blurred Lines EP" by Robin Thicke), and a right sidebar with activity and recommendations. The bottom of the interface shows a playback control bar with a progress slider and play/pause buttons.

Spotify Premium

File Edit View Playback Help

Search

MAIN

- What's New
- Follow
- Inbox
- Play Queue
- Devices

APPS

- App Finder
- Top Lists
- Radio
- Mr. Quizter
- Nick Cave

COLLECTION

- Library
- Local Files
- Starred
- New Playlist
- Star Wars Cantina Band
- Daft Punk - Random Access Memories
- Daft Punk - Get Lucky

Blurred Lines EP

Robin Thicke

Top playlists near you

- Digster HITS by Digster Sverige
- Digster FRESH! by Digster Sverige
- DANCEFLOOR HITS by Filtr Sweden
- Digster DANCEFLOOR by Digster Sverige
- HITS JUST NU! by Filtr Sweden

Top tracks in your network

Title	Artist
Gentleman	Psy
Only Teardrops	Emmelie de Forest
Get Lucky - Radio Edit	Daft Punk, Pharrell ...
Can't Hold Us - feat. Ray ...	Macklemore & Ryan...
Within	Daft Punk
Thrift Shop - feat. Wanz	Macklemore & Ryan...
Lose Yourself to Dance	Daft Punk, Pharrell ...
Get Lucky	Daft Punk, Pharrell ...

People you may find interesting

- Justice
- Lykke Li
- Christina Perri

Who to Follow

- Frida Hoas (5 followers)
- Thrift Shop - feat. Wanz (Macklemore & Ryan Lewis)
- You and Your Sister (Re... (This Mortal Coil)
- Contact (Daft Punk)
- Lie Down Here (& Be ... (Nick Cave & The Bad Seeds)
- Aldrig Vart i Malmö (Maskinen)
- Mass (Good Morning Spider)

Activity

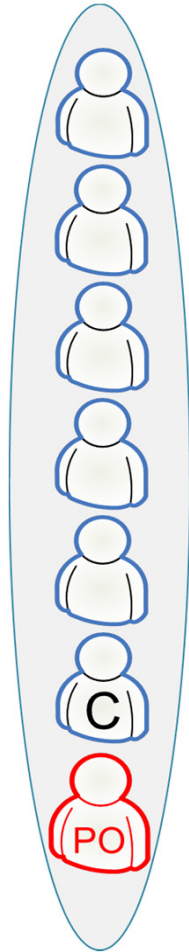
Jimmy Janlén

Touch

Daft Punk - Paul Williams



# Squad

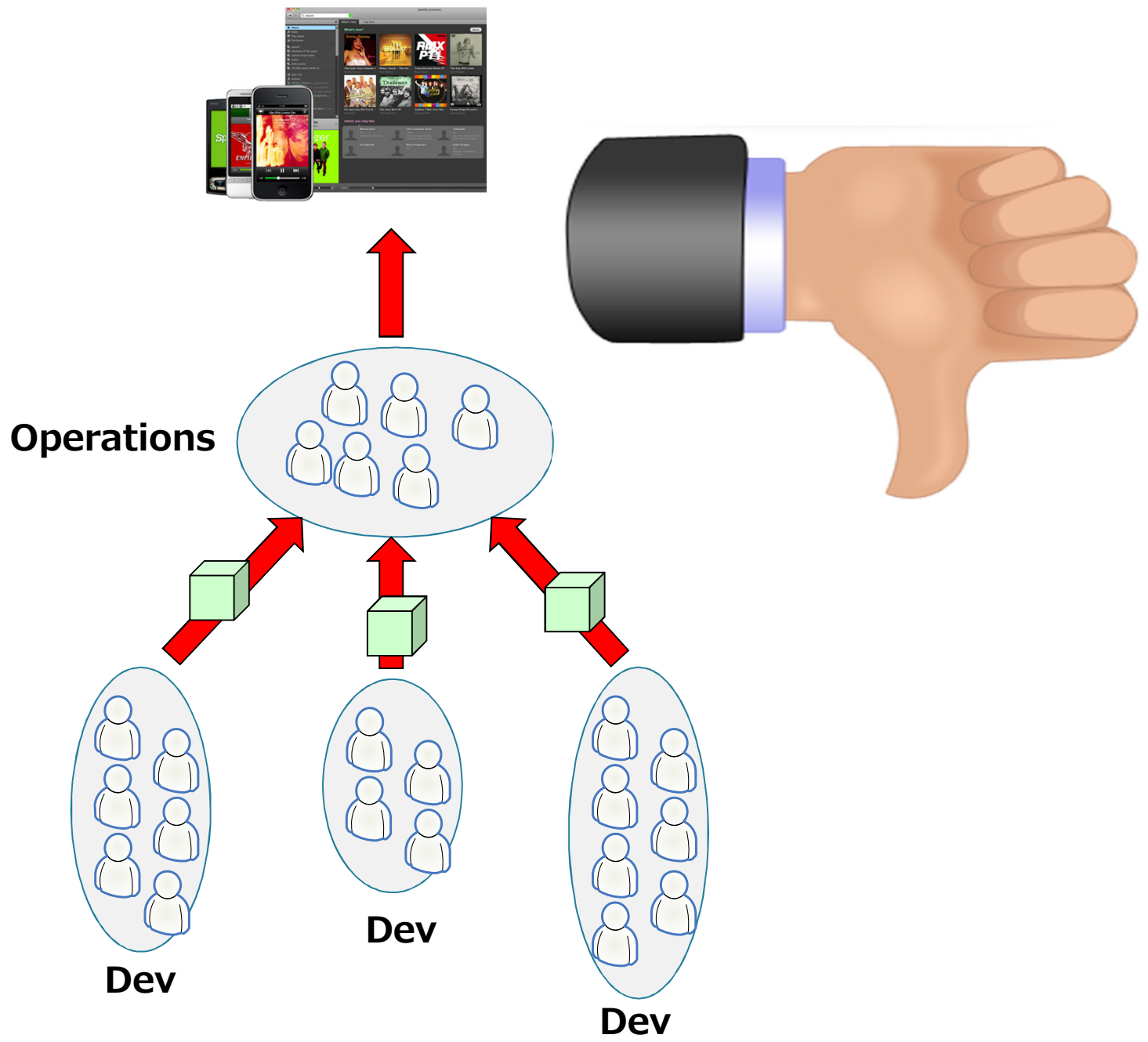


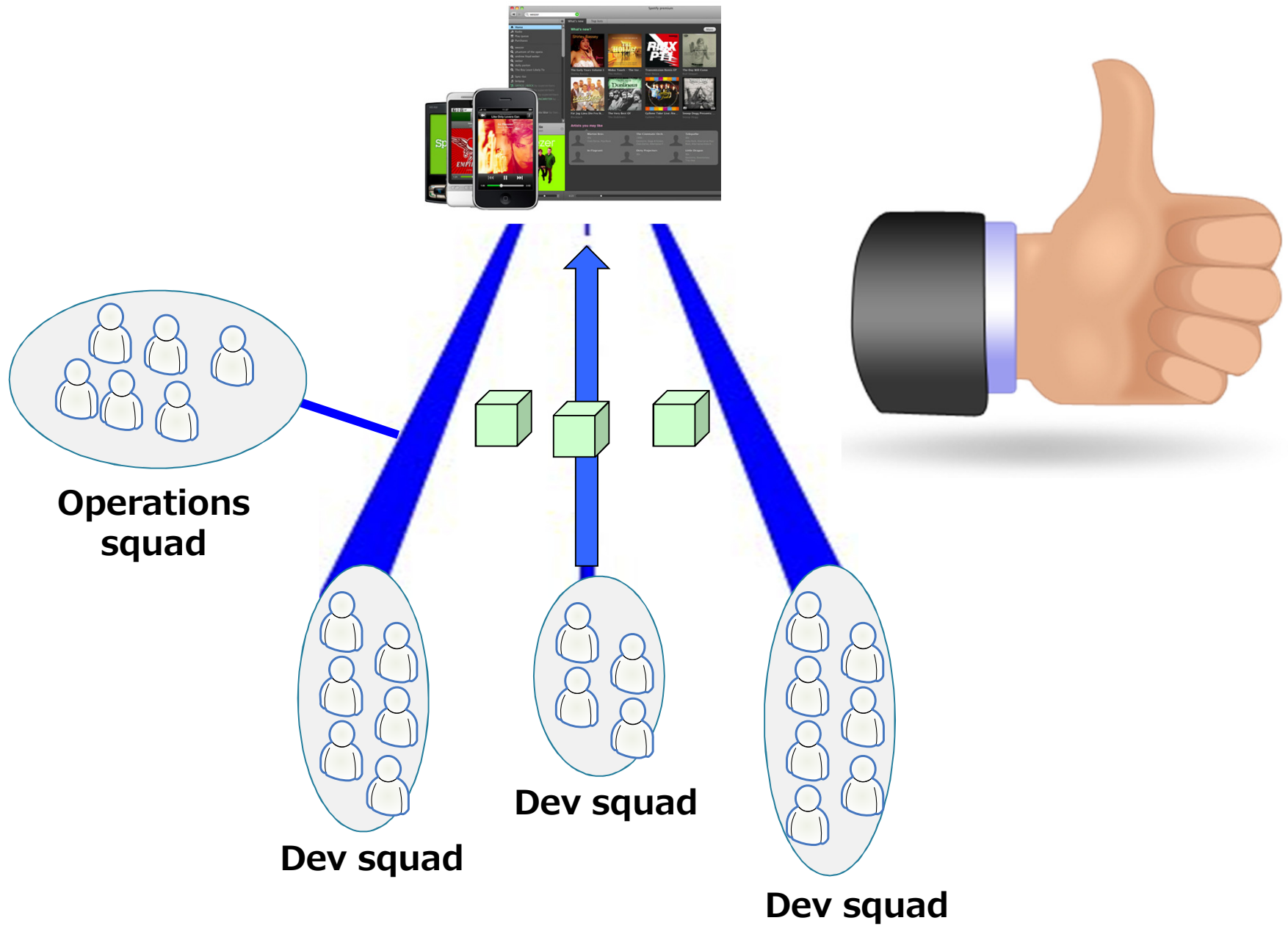
- ✓ **“Feel like a mini-startup”**
- ✓ **Self-organizing**
- ✓ **Cross-functional**
- ✓ **5-7 engineers, less than 10**
- ✓ **Stable**
  
- ✓ **Owns process and quality**
- ✓ **Delivers to prod**

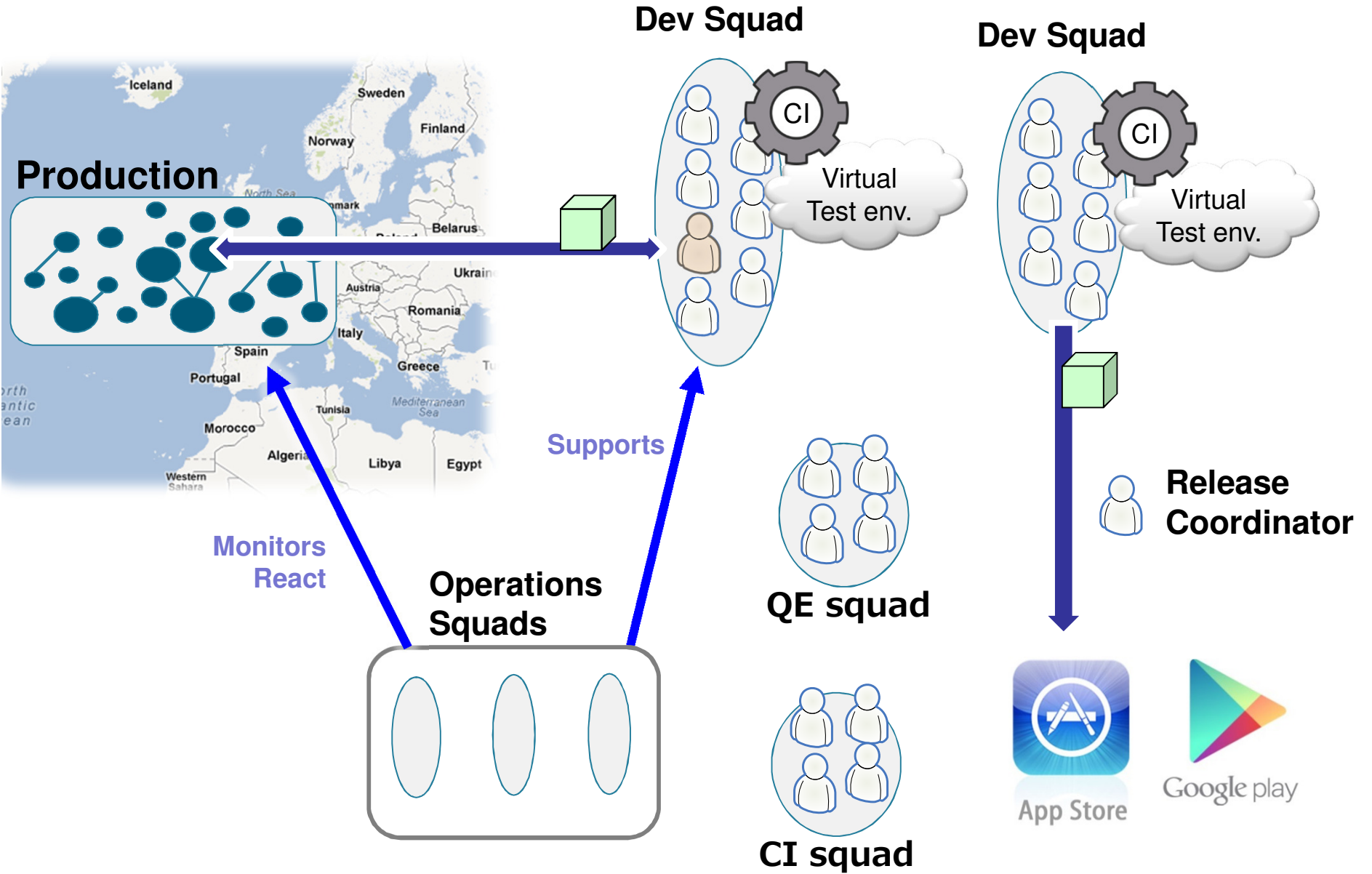


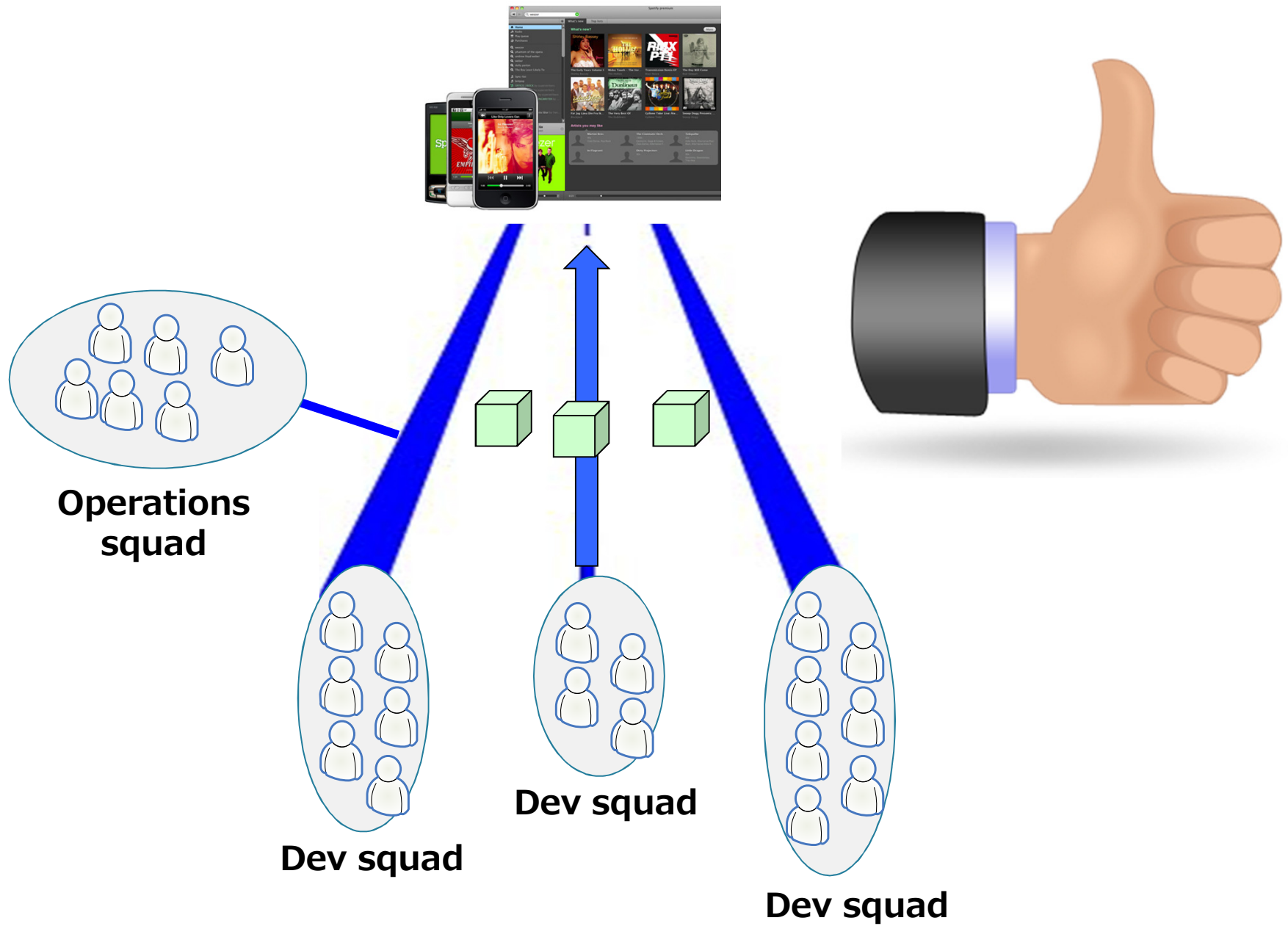
# From Squad to Prod?













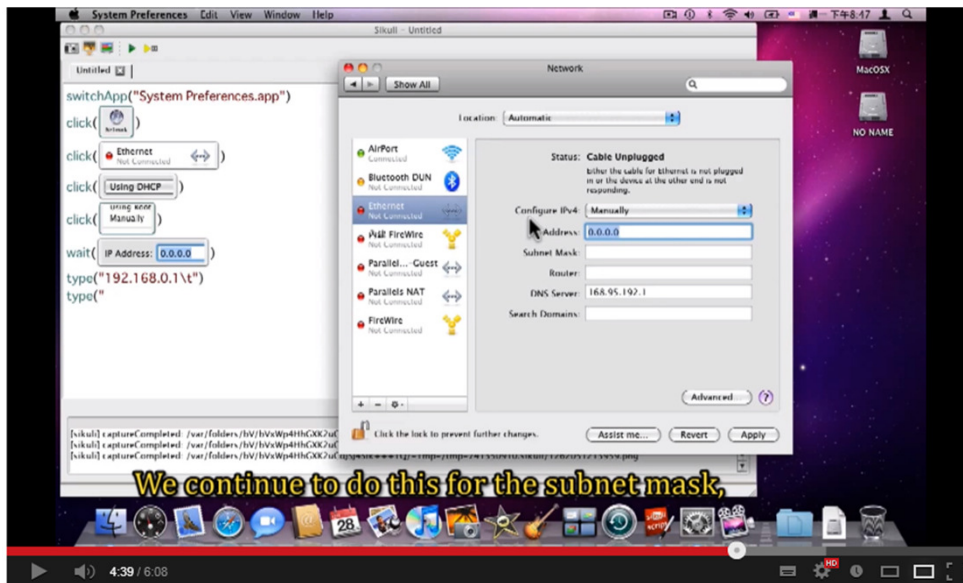
# Non-test Automation

Make room for speed and  
valuable testing!

# Automate more!



# Automate more!



*SIKULI – Automate (and test) graphical UI interfaces*

Automate to free up time and to speed up!

Simplify...

- Environment preparations
- Installation routines
- Release packaging
- Test Data preparation
- Reports
- Branch merging
- Repetitive (and boring) tasks
- Etc.

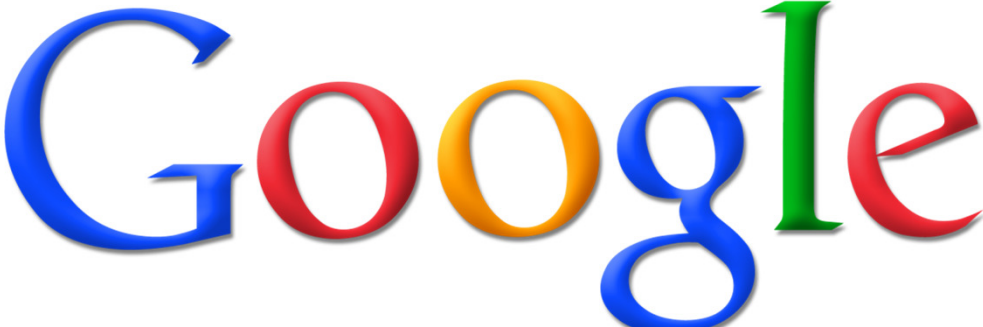




# Testing as a service

# Blocket

The screenshot shows the Blocket.se website in a browser window. The address bar displays 'www.blocket.se'. The page features a navigation bar with news snippets: 'SPORT: Kazakiska vindar kring Johansson', 'NÄRINGSLIV: Apple lanserar nya modeller av Iphone 5', and 'VÄDRET: 16 °C Stockholm'. The main logo 'blocket.se' is prominently displayed with the tagline 'Sveriges största Köp & Sälj marknad'. Below the logo are three main categories: 'bostad' (Real Estate), 'jobb' (Jobs), and 'shopping'. A featured job advertisement titled 'Nytt jobb efter semestern?' lists various job categories and their counts. To the right, a map of Sweden is shown with a list of regions for selection. At the bottom, there are sections for 'annons' (advertisements) and 'Nytt sedan sist!' (New arrivals) featuring car listings.

The Google logo is displayed in its characteristic multi-colored font (blue, red, yellow, blue, green, red) with a slight 3D effect. It is centered within a white rectangular area that has a thick, dark blue border.

Google

**ONE branch!**

# Google



Everyone works on the master branch

All code tested continuously



Release every 6<sup>th</sup> week

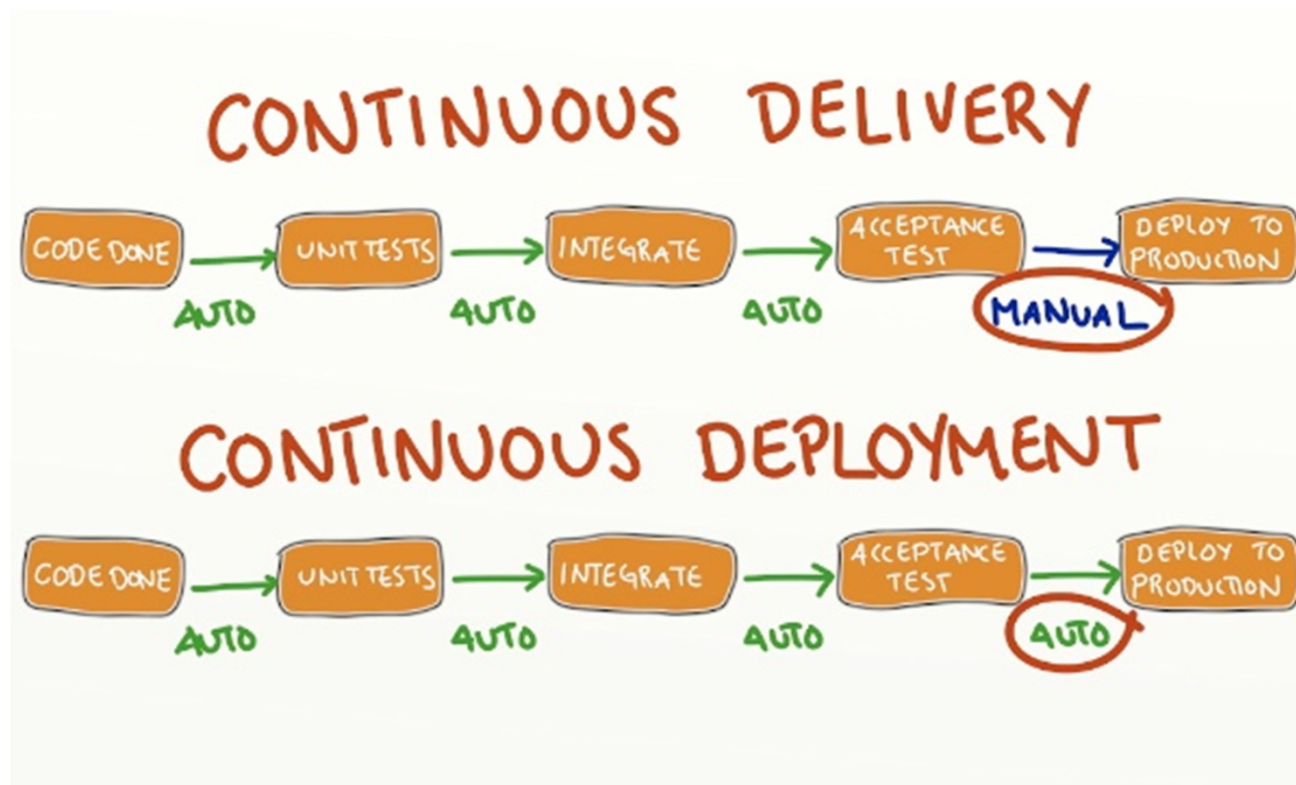
Failed Important Build -  
"Revert first, ask questions later"

+ Testers in every team

# Current Trends & The Future

# Release Pipeline

# Continuous Delivery vs Continuous Deployment



# Dark launching



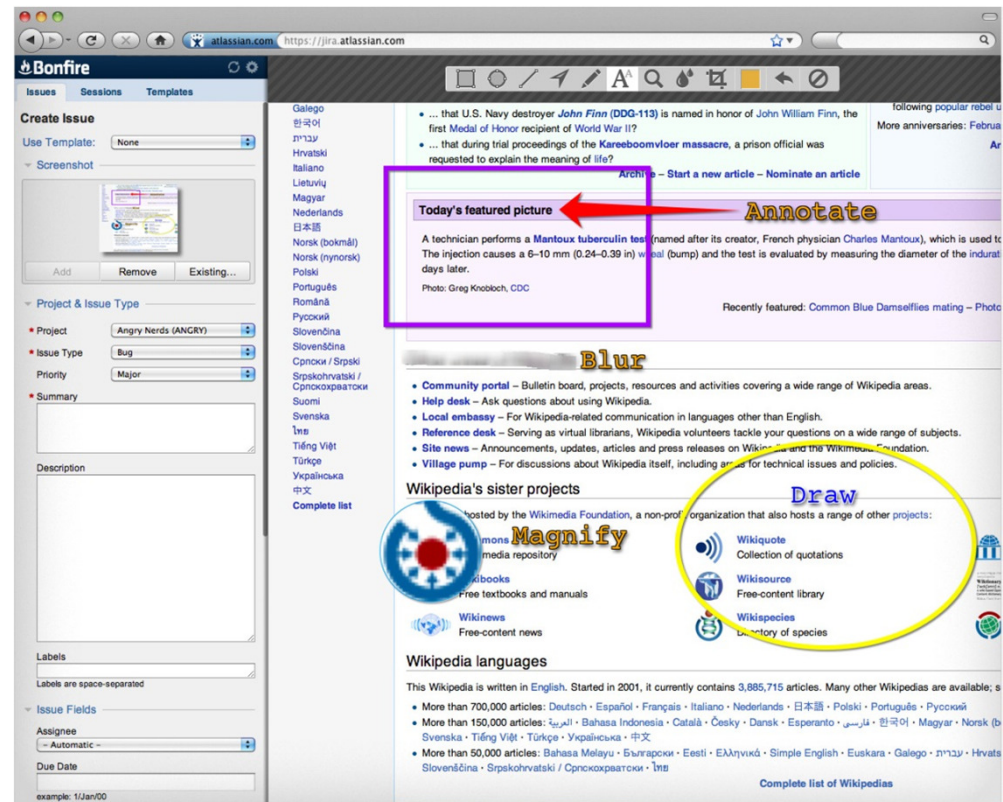
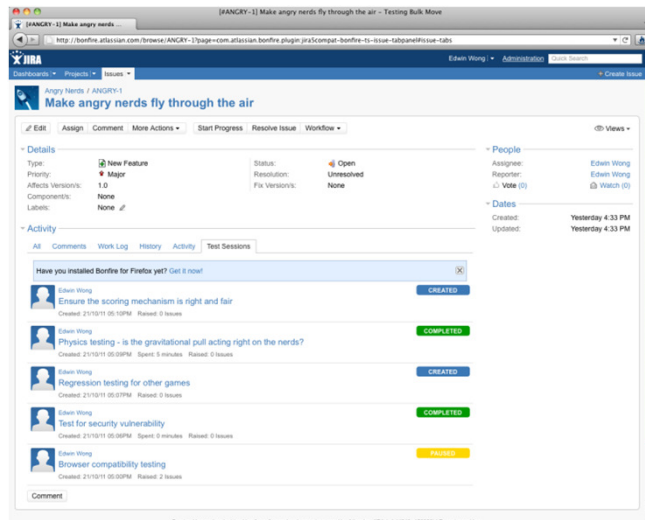


# Automatic Rollback

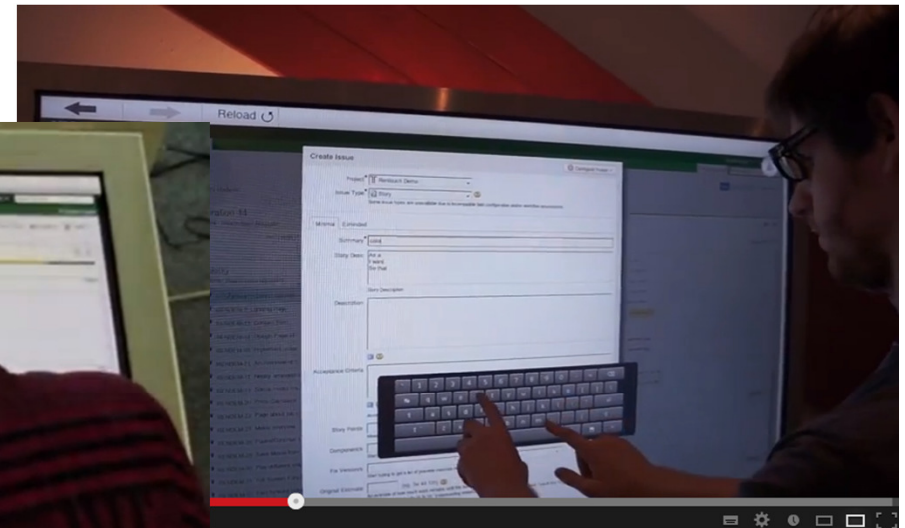
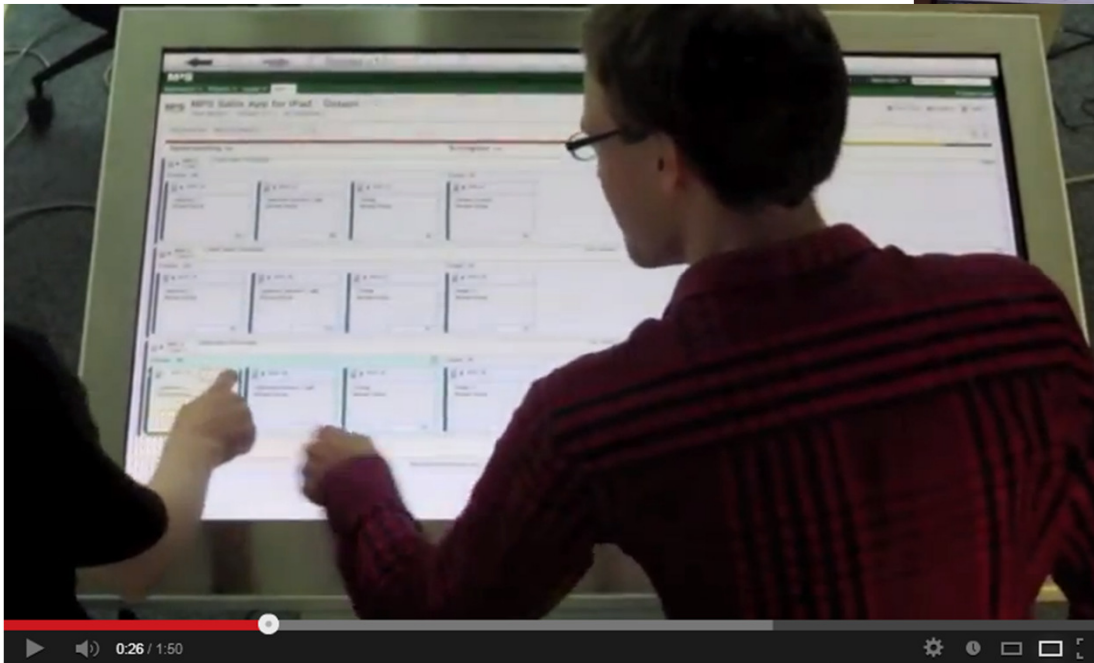


# Tools

# Tool integrations

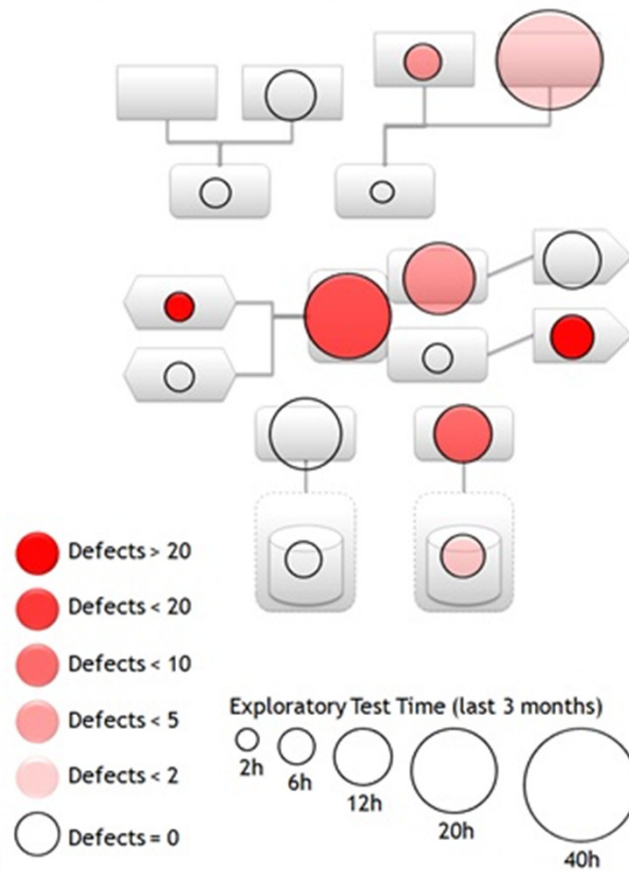


# Digital Scrum Dashboards



# Visualizations

## Exploratory Heat Map





*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



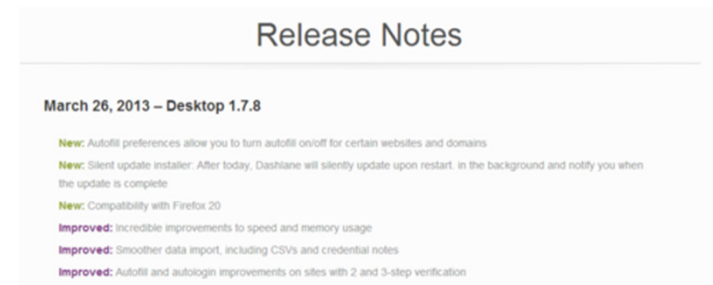
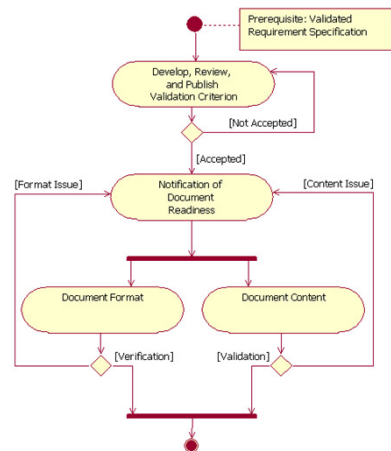
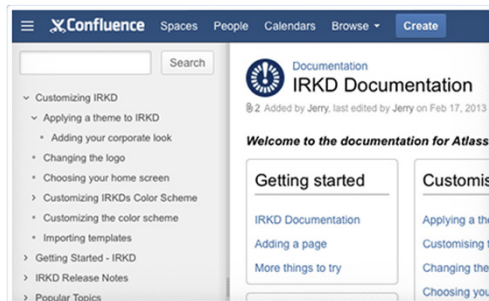
# Atlas Copco – A-TDD Example

## 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  
will kill the tester

# Myth: Automation will kill the tester?



## BUSTED

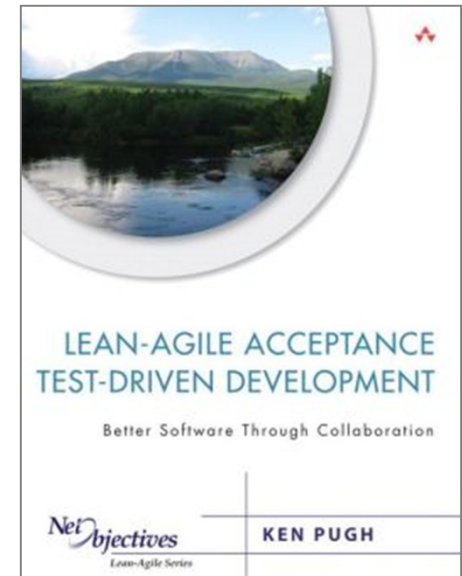
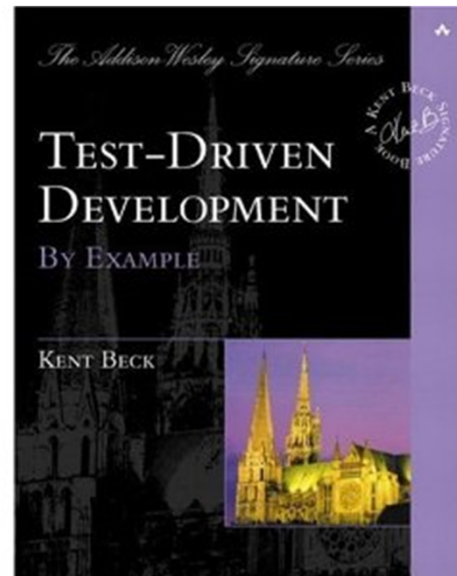
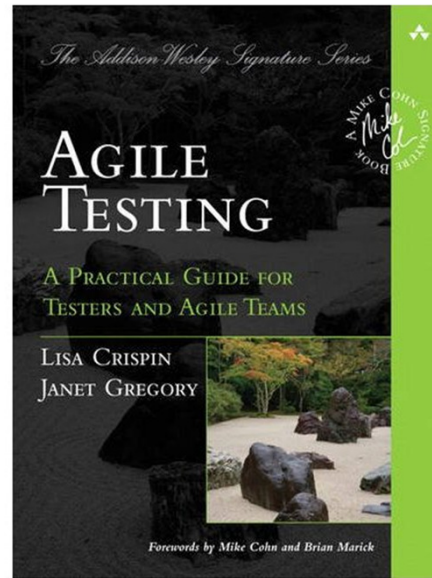
- Skill still needed, broadens the team (XFT, end-to-end)
- Tester → QA/QE/TA/QC/etc.
- Strengthens the team's planning, design and ability to deliver with their input, expertise and experience
- Broadens the perspective (process, testability, etc.)

Thank you!





Reading...





# 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](mailto:jimmy.janlen@gmail.com)



**YouTube: JimmyJanlen**  
<http://www.youtube.com/user/JimmyJanlen>



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



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



**Jimmy**

**Janlén**

98



**Jimmy Janlén**