



HOW TO BE PROUD
WHEN YOU ARE DONE

Mikalai Alimenkou

- Java Technical Lead/Scrum Master at **Zoral Labs**
- 6+ years in software development
- 4+ years of working by Agile methodologies
- **Expert** in Agile engineering practices
- Agile coach at **XP Injection**



Aleksey Solntsev

- Architect at **Infopulse Ukraine**
- Agile volunteer
- **Certified Scrum Practitioner**
- Coordinator of translation of the book "**Scrum and XP from the Trenches**" and "**Kanban and Scrum – making the most of both**" into Russian
- Agile coach at **XP Injection**

Everybody has own “dreams”

- ✓ Latest frameworks
- ✓ No overtimes
- ✓ No bug fixing
- ✓ Clean code



Team

- ✓ More features
- ✓ In time delivery
- ✓ No defects!



Customer

- ✓ Goal achievement
- ✓ Predictable plans
- ✓ On budget



Manager

What all of them have in common?

Done product



With fast delivery



At low price



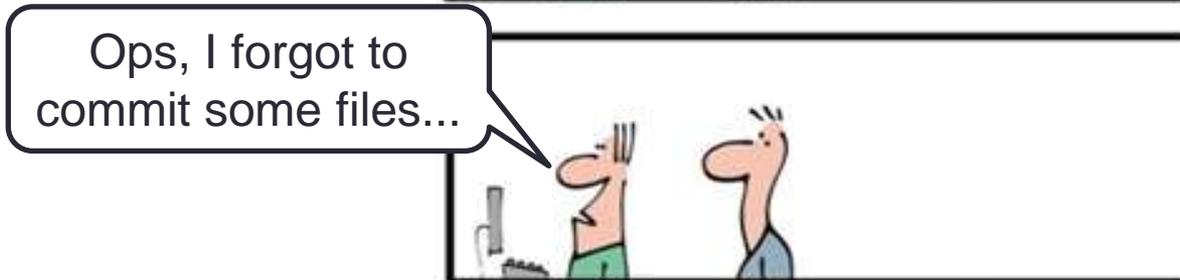
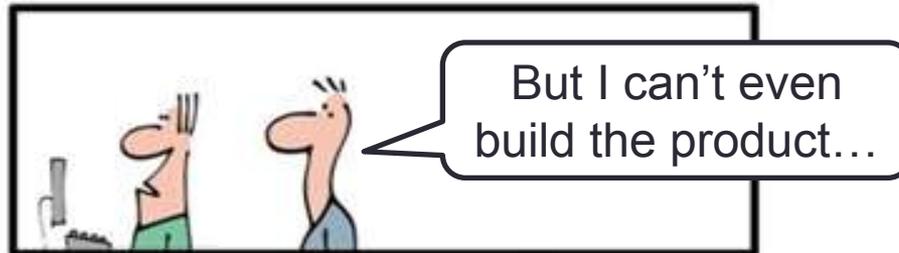
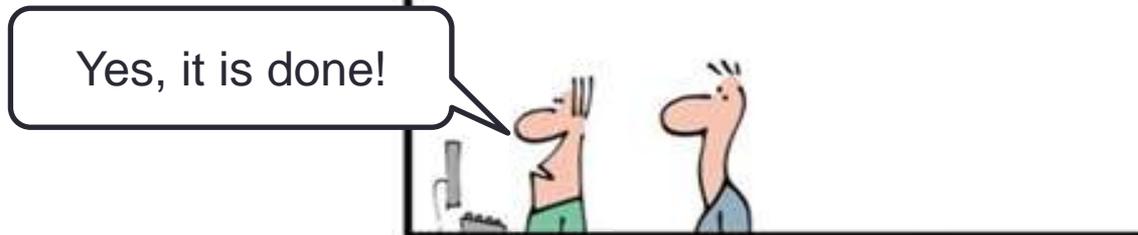
And good quality



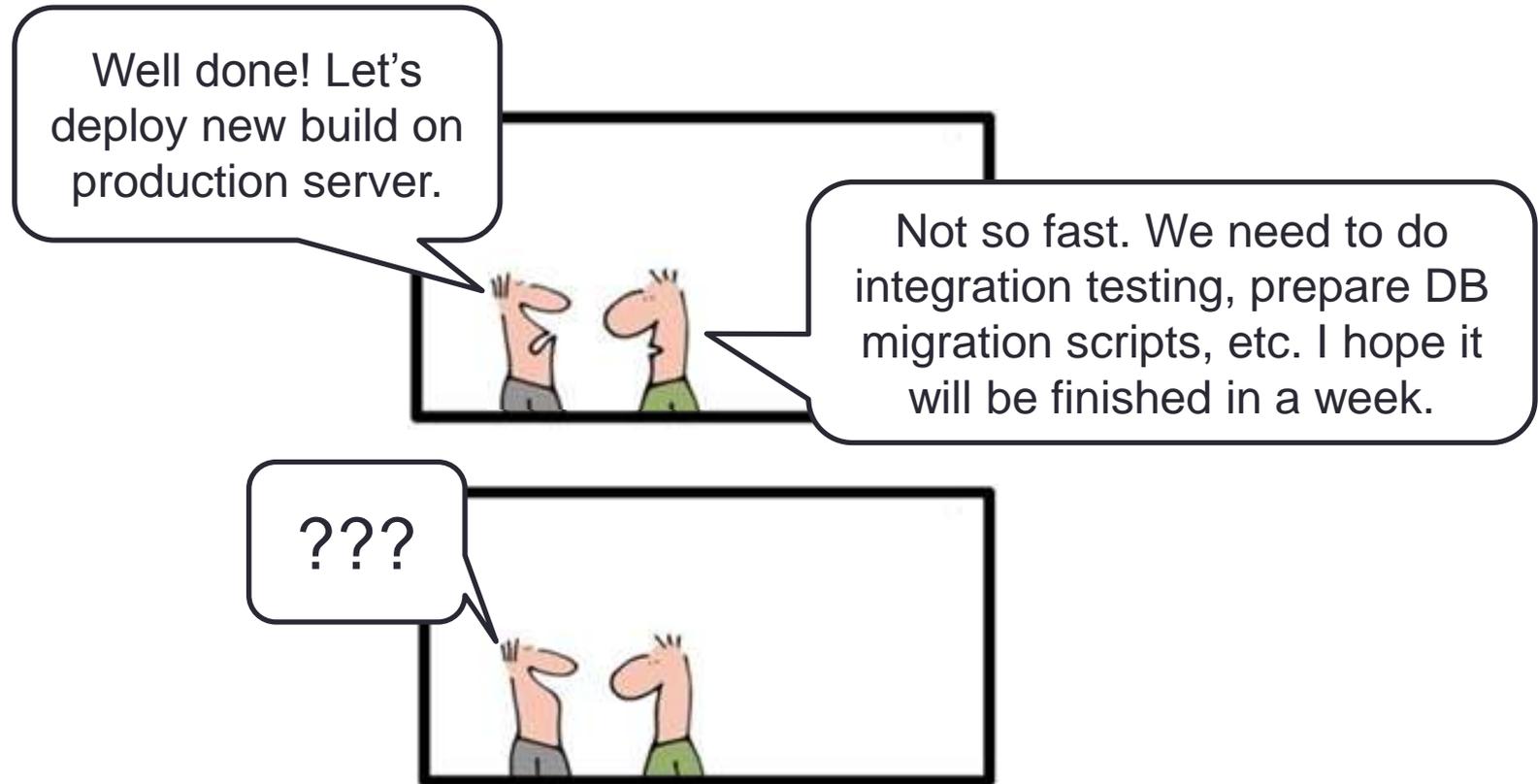
Why in real life it is not so simple?



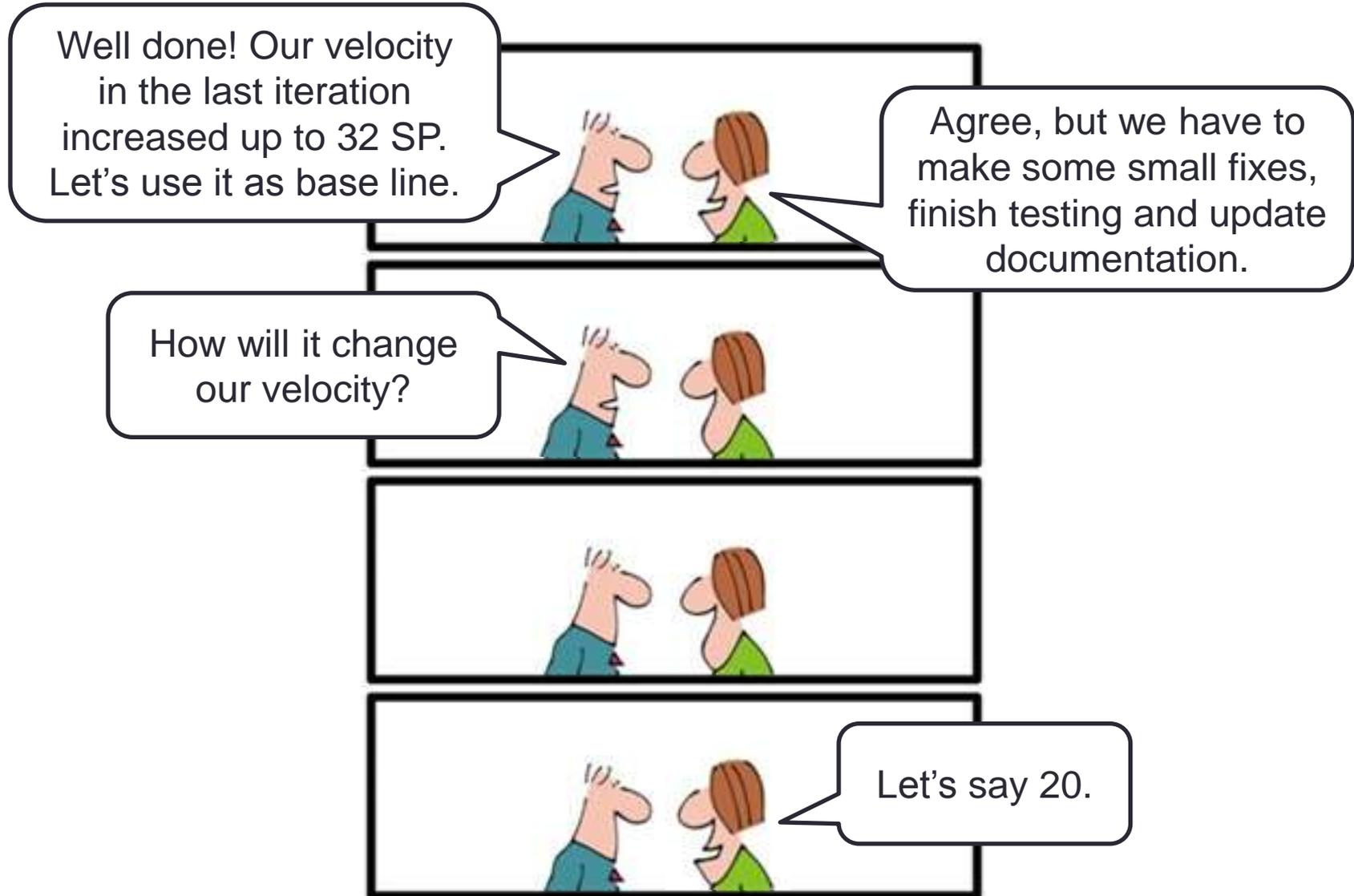
“Uncommitted stuff”



“Useless build”



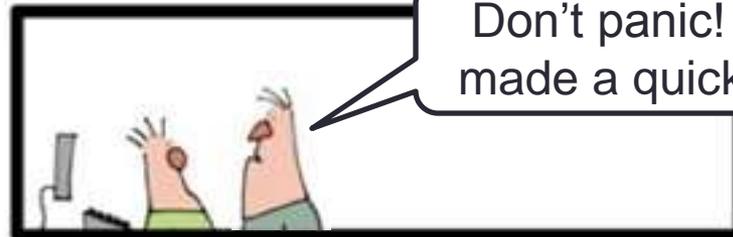
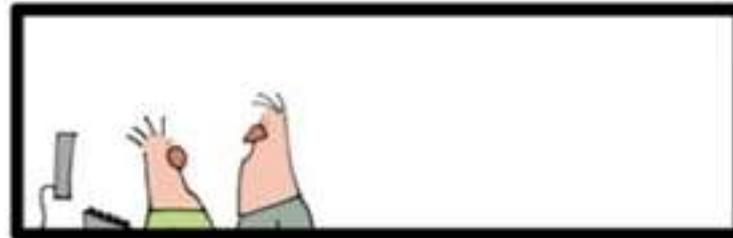
“Unstable velocity”



“Unverified tasks”

WTF! A new order form is nightmare. It looks like random set of fields and doesn't accept basic values.

Have somebody tried to use it before assign to me?

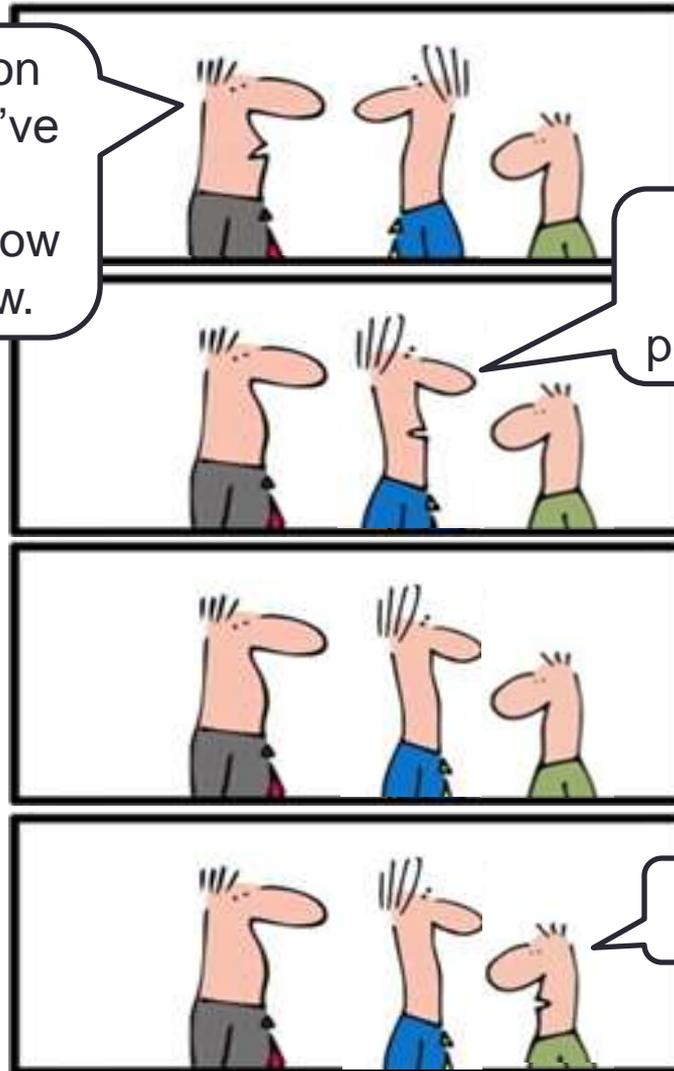


“Forgotten requirements”

Guys, as we discussed on the planning meeting, we've just started pre-orders program for iPad2. And now our site is extremely slow.

John, have you made load and performance testing?

Well, but ...



Why this happens?



Whhhhyyyy?!!!

Hidden conflicts in goals

- **Developers** like always implementing interesting features, but **customers** want working software
- **Team** want to show productivity, but **customers** want predictability
- **Developers** believe they are perfect, but **customers** want software without defects
- **Management** want to deliver in time, but **customers** want ready to use software
- **Developers** see technical side of the project, but **customers** see it from end users perspective

And the main reason is ...

*Everybody has his own definition of
words 'done', 'fast', 'low
price', 'good quality'!*

Let's start from definition

“Definition of Done” is

- ✓ a "contract" between all parties on subject when ...
- ✓ a checklist of valuable activities for ...
- ✓ gates your product has to go through before ...

... you can label the product “Done”



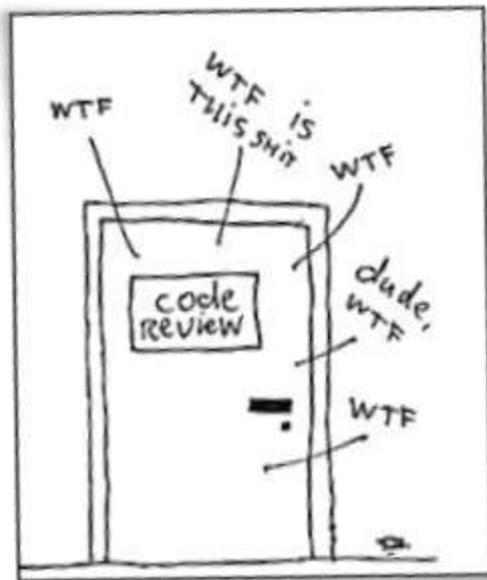
4

main
questions



How to start?

Take from previous project



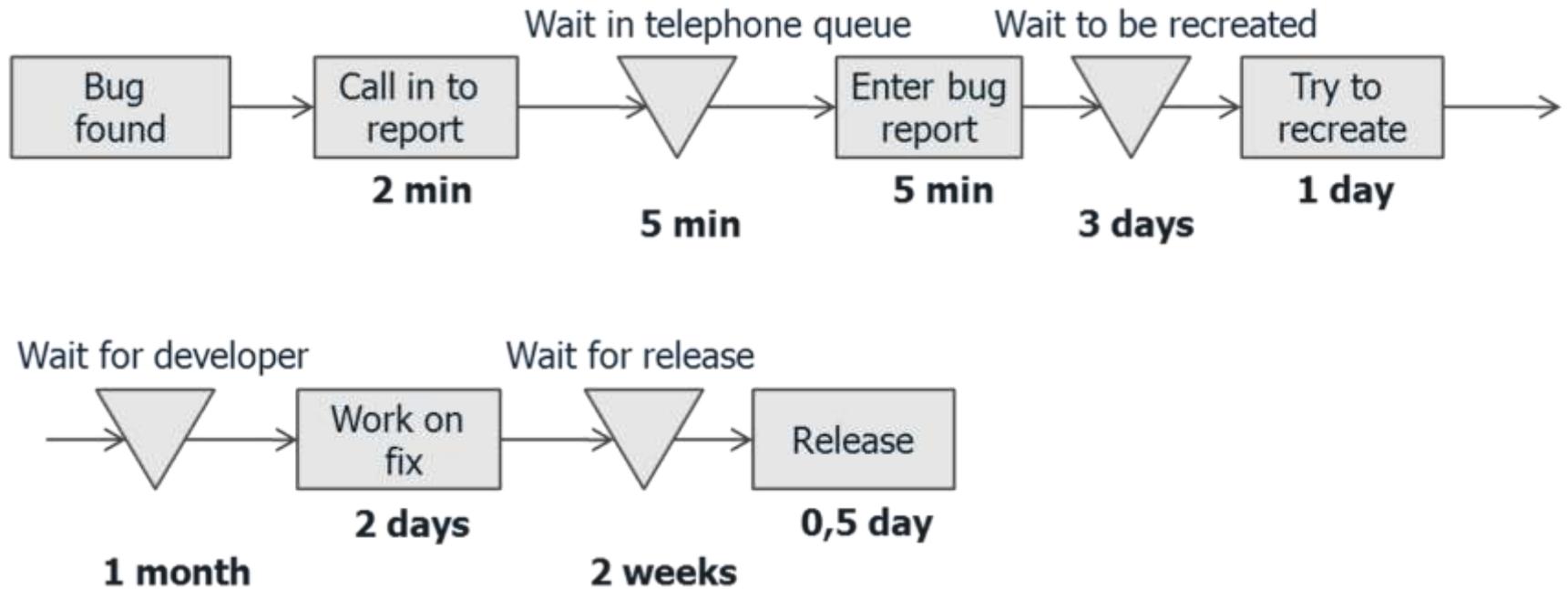
From code quality in mind



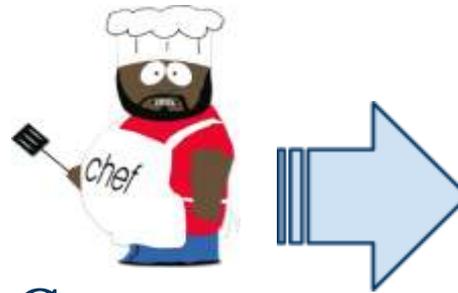
From business problems and wishes



Start from flow visualization



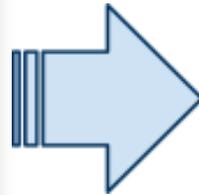
Who should define?



Customer



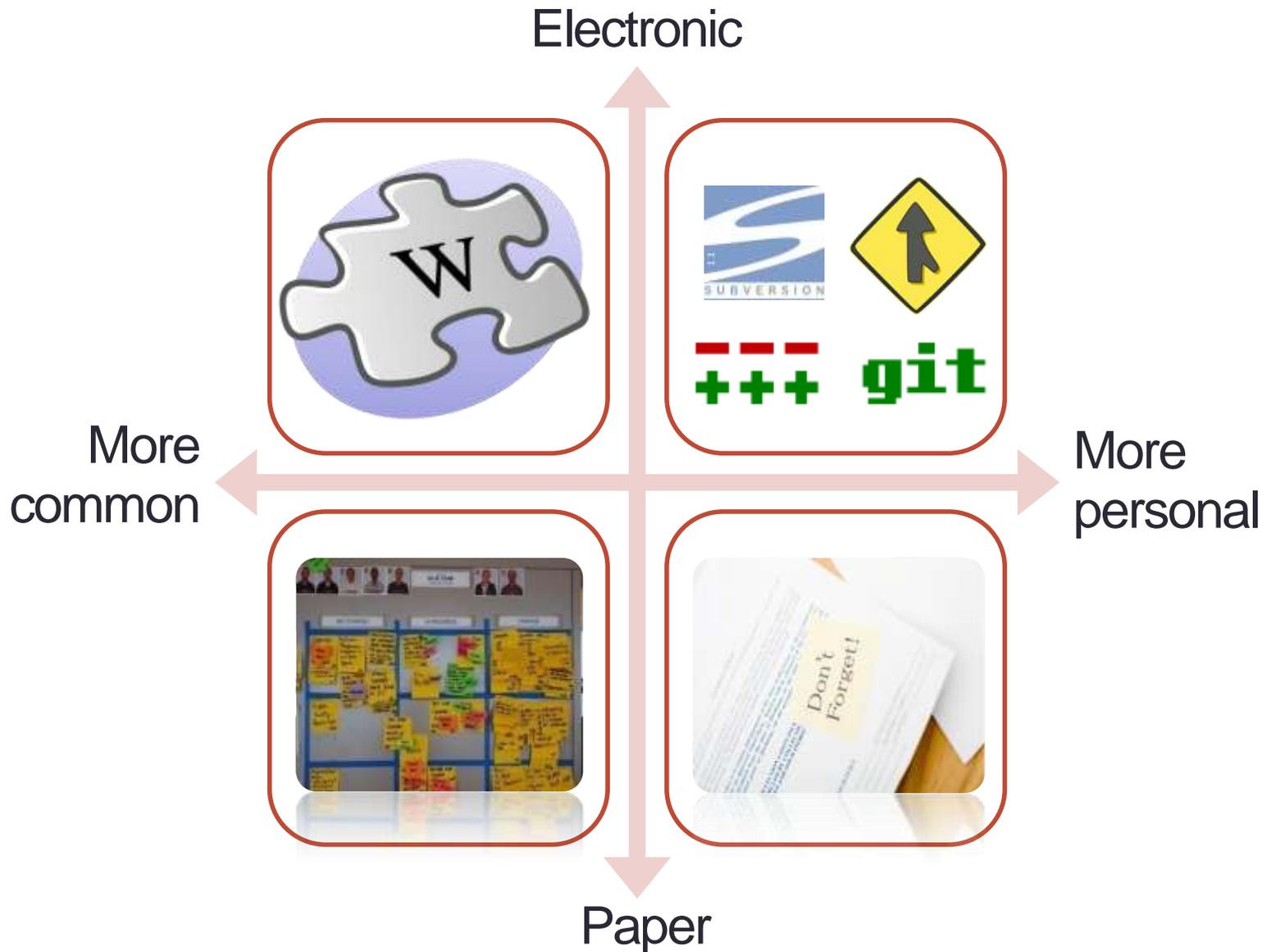
Team



When to define?



Where to store?



3 main formats

 item 1

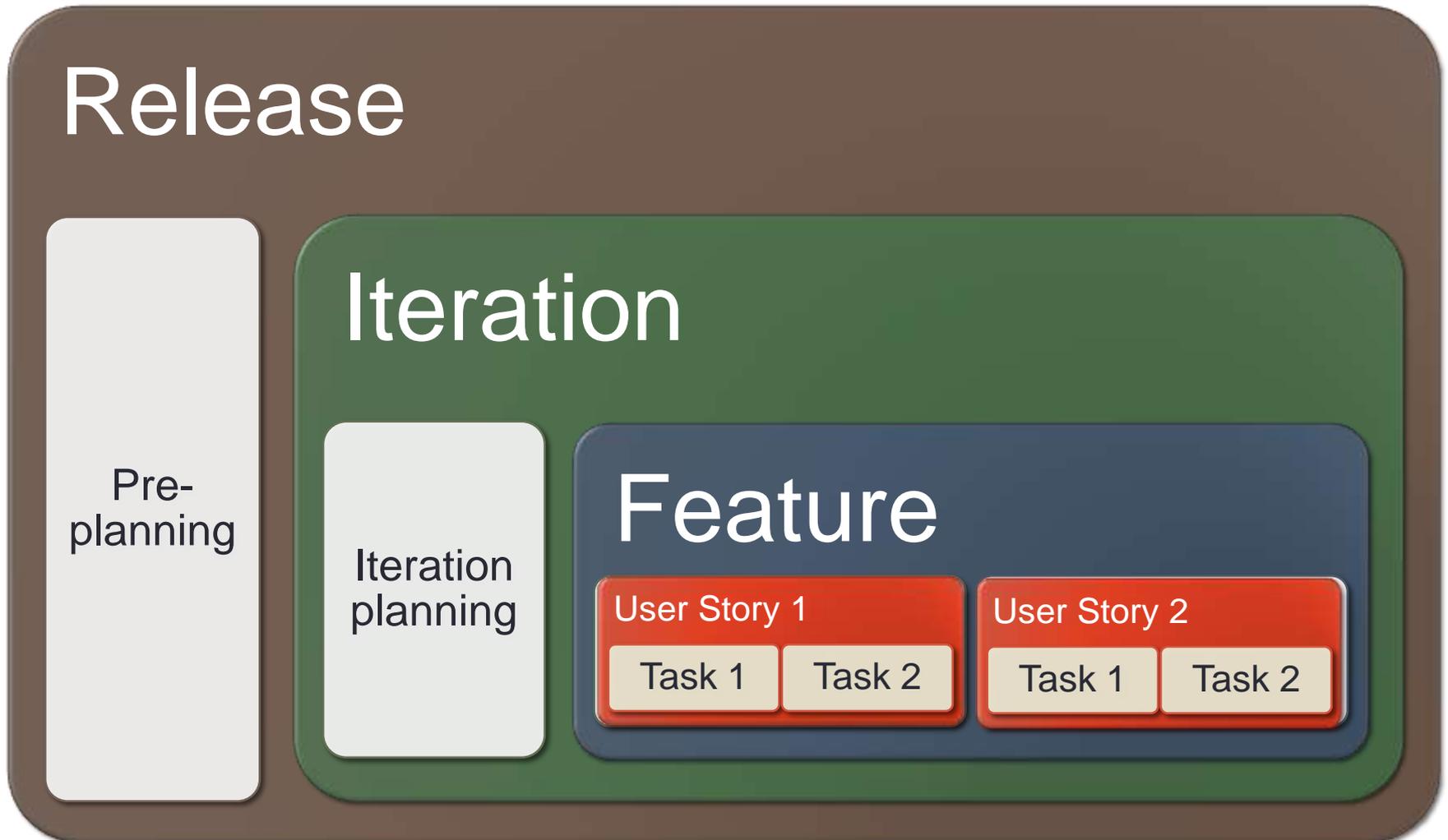
 item 2

Plain list

EXAMPLE: DEFINITION OF DONE

1. CODE IS COMMITTED INTO VCS
2. UNIT TESTS ARE CREATED FOR ALL CODE
3. INTEGRATION TESTS ARE CREATED FOR ALL AFFECTED MODULES
4. AUTOMATED FUNCTIONAL TESTS ARE CREATED
5. ALL TESTS ARE COMMITTED INTO VCS
6. ALL TESTS ARE INCLUDED INTO AUTOMATED BUILD
7. CI BUILD PASSED SUCCESSFULLY
8. STATIC ANALYSIS TOOLS ARE APPLIED ARE UP TO DATE
9. PROJECT CODE METRICS ARE UP TO DATE
10. ANY ISSUES FOUND BY STATIC ANALYSIS TOOLS ARE FIXED
11. CODE IS REVIEWED BY AT LEAST ONE OTHER TEAM MEMBER
12. QA TESTED APPLICATION WITH NEW CODE TO VERIFY NEW FUNCTIONALITY
13. REGRESSION TESTING IS PERFORMED BY QA TEAM
14. NEW CODE IS DEPLOYED ON THE PRODUCTION ENVIRONMENT
15. NEW FUNCTIONALITY IS ACCEPTED BY CUSTOMER

Different levels of granularity



List categorized by level

Pre-planning

- Criteria 1
- ...
- Criteria N

Iteration

- Criteria 1
- ...
- Criteria N

Feature

- Criteria 1
- ...
- Criteria N

User Story

- Criteria 1
- ...
- Criteria N

Task

- Criteria 1
- ...
- Criteria N

Release

- Criteria 1
- ...
- Criteria N

Complex structure

Pre-planning

User experience	
Test	Has QA been part of preplanning?

Iteration

User experience	
Test	
Documentation	

Feature

User experience	
Test	
Documentation	

User Story

User experience	
Documentation	Has the end-user documentation been updated?

Task

Working software	Has the implementation been reviewed by a peer?
Test	

Release

User experience	
Test	Is the scalability testing done?
Documentation	

3 main control principles



Automation



Use SVN hook for verify comments in commits

A screenshot of a build system interface, likely Jenkins, showing the results of a code inspection for build #24. The interface is titled "Build #24" and includes navigation links for "Home", "BuildServer :: Inspections", "Results", "Build Log", "Changes (11)", and "Code Inspection". The main content area displays the following information:

- Result:** Build has too many problems ... [Run](#)
- Responsible:** No one || [take responsibility](#)
- Time:** 10 Aug 19:43 - 20:11 (28m:16s)
- Agent:** BuildUnit

Below this, there are tabs for "Problems: All (1558)" and "Errors (25)", with "Selected: Deprecated API usage (6)" highlighted. The list of problems includes:

- Declaration has problems in javadoc references
 - [Deprecated API usage \(6\)](#)
 - [equals\(\) and hashCode\(\) not paired \(4\)](#)
- Assignment issues
 - Assignment to method parameter (18)
- Class structure
 - 'public' constructor in non-'public' class (13)
 - 'static', non-'final' field (25)
 - Class may be interface (1)
 - Class without package statement (2)
 - Empty class (1)
 - Field can be local (8)
 - Utility class with public constructor (1)
 - Utility class without private constructor (26)
- Cloning issues
 - Cloneable class without 'clone()' (2)

On the right side, a tree view shows the file structure with the following files and their associated issues:

- build-server4idea/src/jetbrains/buildServer/
 - ExportedTextEditorHighlightingPass.java (1)
 - 57: 'TextEditorHighlightingPass(com.intellij.oper
 - InspectionPassRegistrar.java (2)
 - 65:
 - 'registerTextEditorHighlightingPass(com.intellij.c
 - int, int)' is deprecated
 - 65: 'LAST' is deprecated
- web-test/src/jetbrains/buildServer/control
 - MockSession.java (2)
- xml-rpc-wrapper/src/org/apache/xmlrpc
 - TCDefaultXmlRpcTransport.java (1)

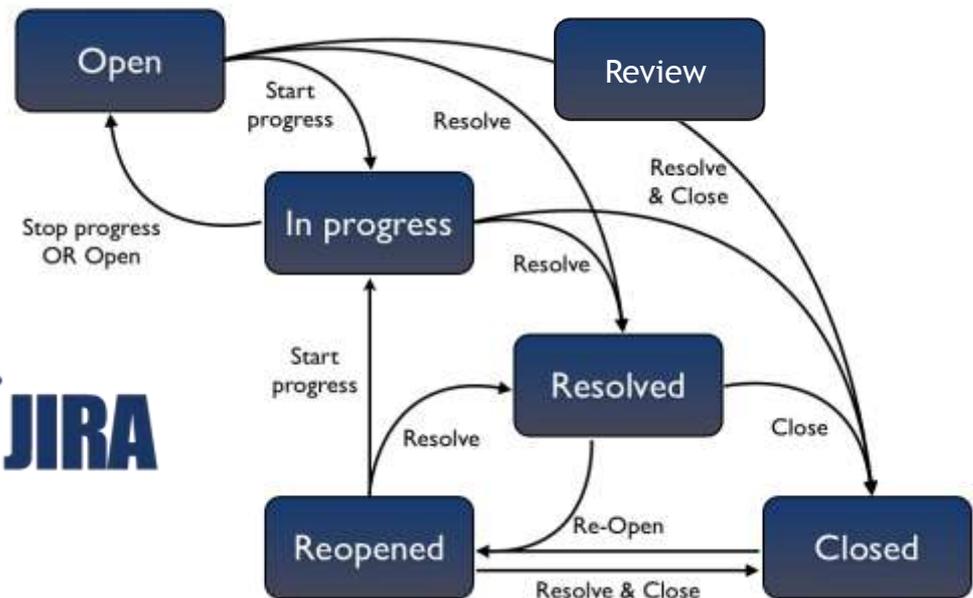
Add static analyzer to build

Fixed workflow



Fix and document workflow

Task tracking system can help



Responsible persons

Team Lead

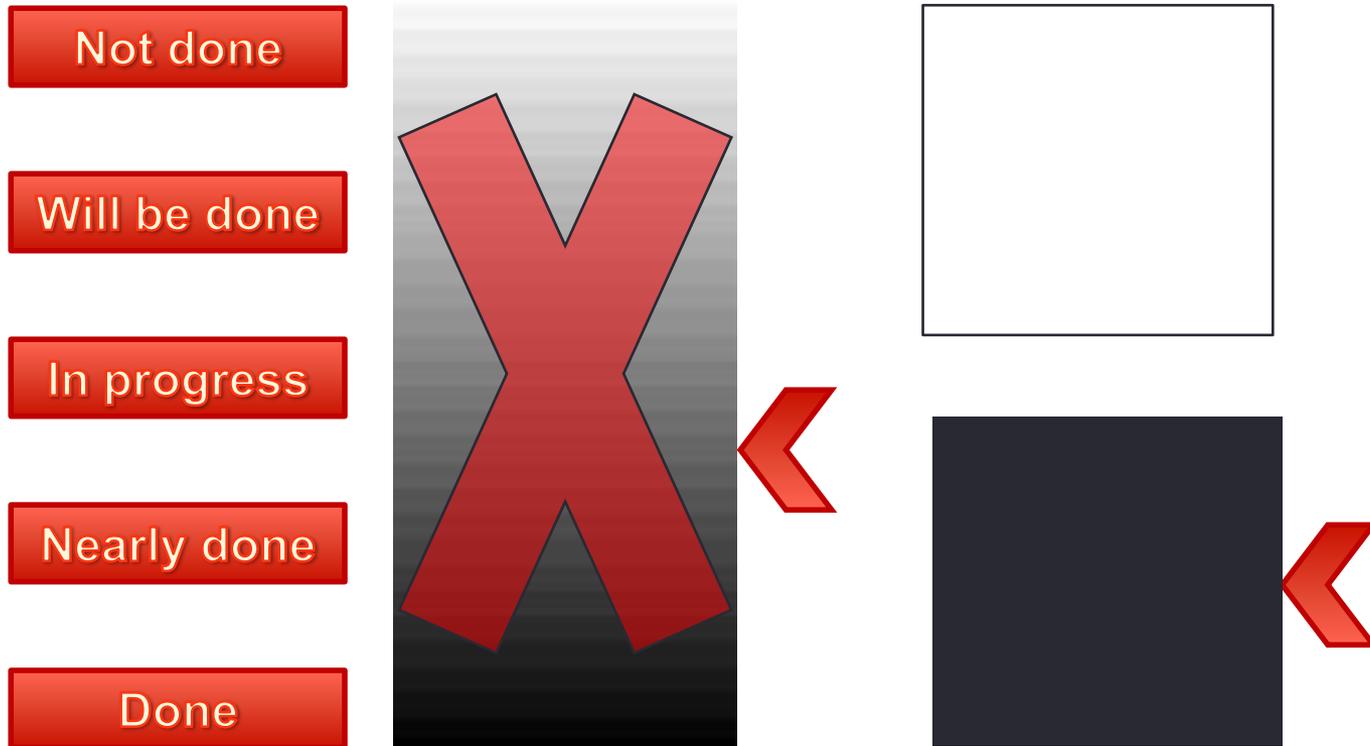
Manager



How does it work
in real world

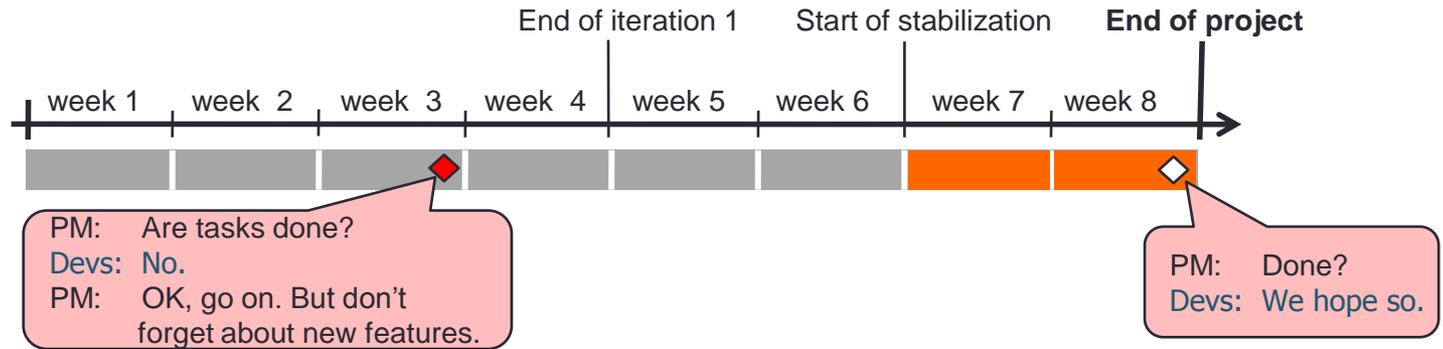


Binary state

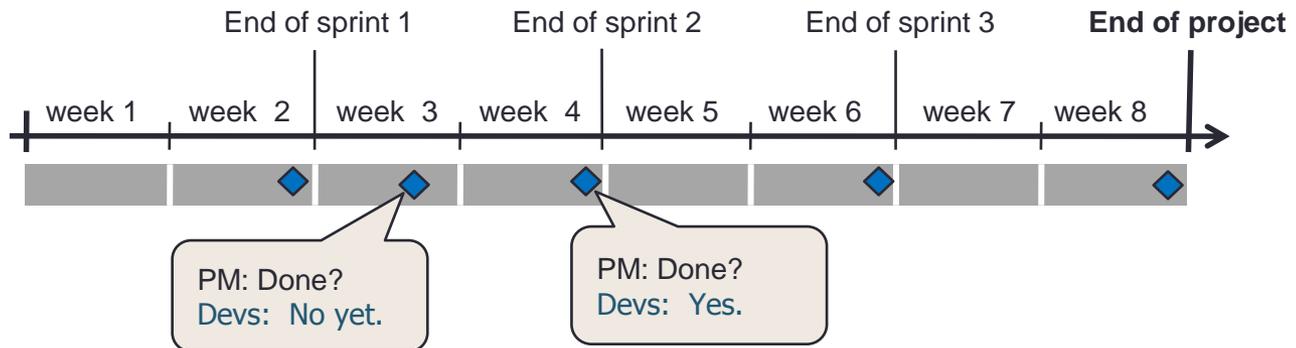


Depends on context

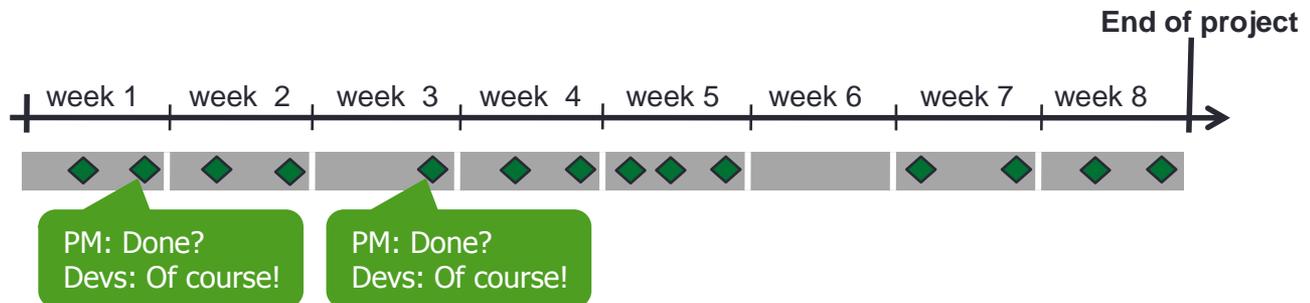
Traditional approaches



Scrum



Kanban



7

main
problems



No common understanding



No commitment



Unrealistic



Too ideal



Partially done tasks are accepted



"Broken windows" principle



“Development” only



Let's try to build Definition of Done!



Conclusions

Common vocabulary helps us to avoid hidden conflicts ...



... and work together as a team to archive our goals!

- Introduce “Definition of Done” ASAP to avoid broken expectations
- All parties must take part in definition process
- Automate as much as possible
- Don't lie yourself, DONE means DONE
- Learn from your mistakes
- Inspect and adapt continuously

**Take it
away**

Q&A



Email us:

- mikalai.alimenkou@xpinjection.com
- aleksey.solntsev@xpinjection.com

