# Agile Software Development

David Branner

20151211

## What is it?

1. A theory about efficient software development
2. A movement based on the theory (jargony)
3. A set of practices – my focus today

My aim: to present jargon and practice together.

The prime issue is how teams of developers can work most efficiently.

## Key high-level ideas

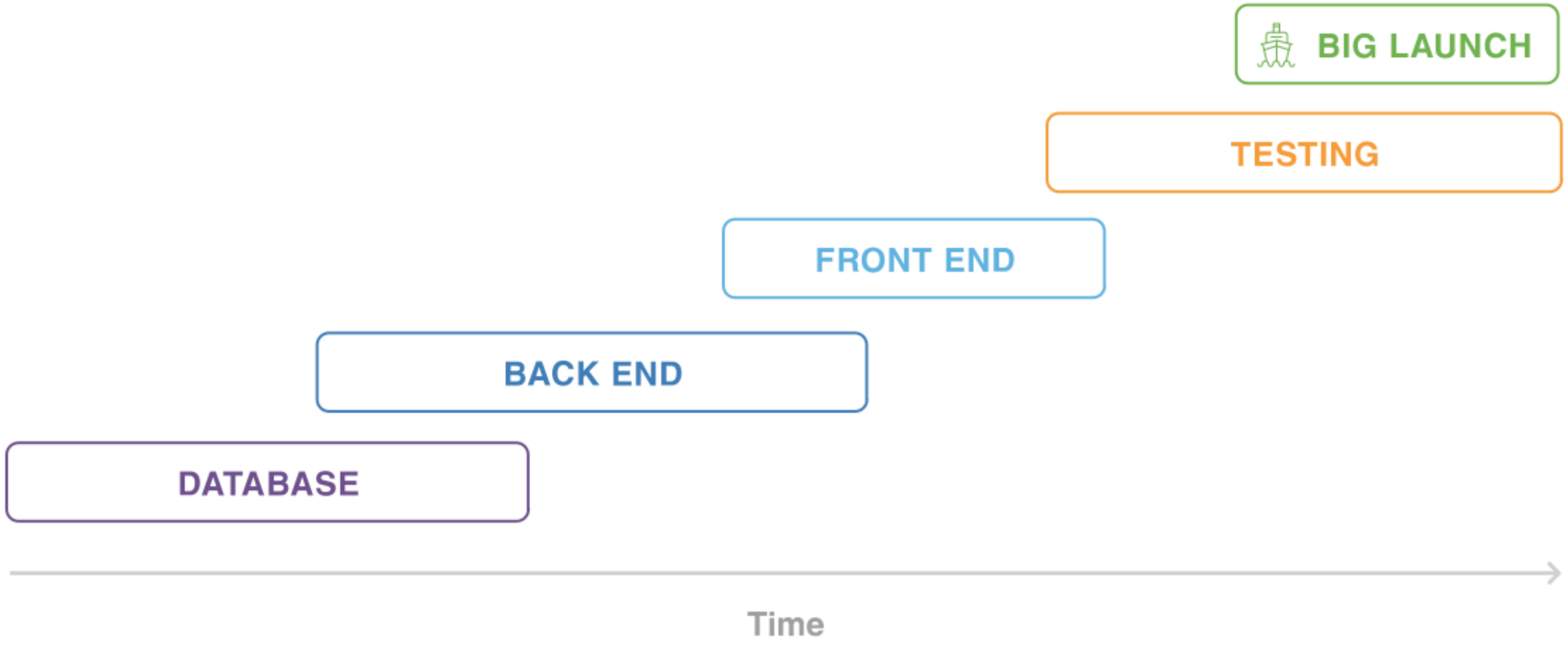
1. modular — work on components independently
2. iterative — revise components over and over
3. short dev. cycle* — incremental deliverables

Modular: like separation of concerns in code and asynchronous callbacks in code.

Advocates contrast Agile with "waterfall"* model.

# (Digression: “waterfall” model – 1)

1. characterized by relatively rigid sequence
2. not always bad – good for material production



BIG LAUNCH

TESTING

FRONT END

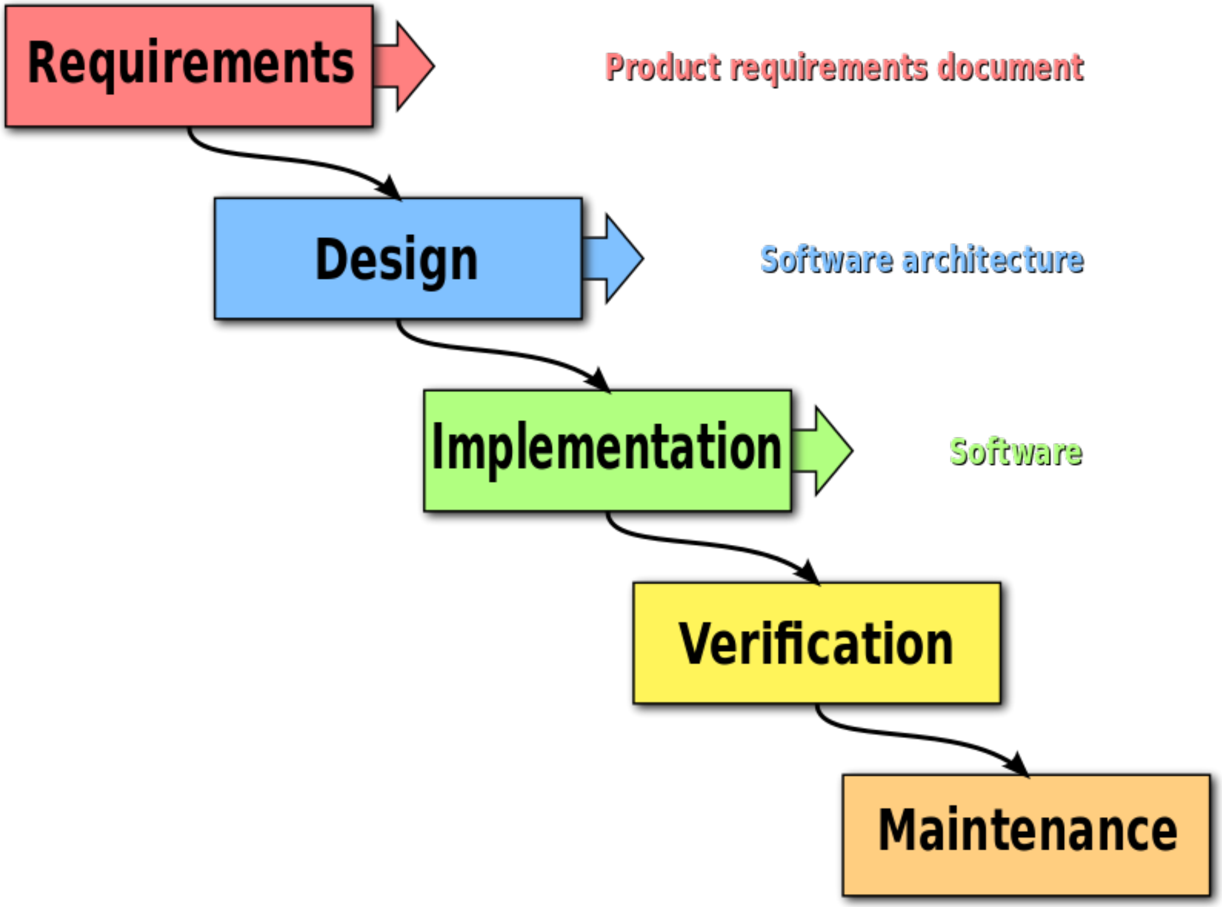
BACK END

DATABASE

Time

(one vision, from Atlassian’s site)

# (Digression: “waterfall” model – 2)



(another vision, from en.Wikipedia.org)

# In contrast: Agile development cycle: “sprint”

1. fixed length (short) but repeats over and over
2. managed with a process called “scrum”*
3. always supposed to end with a new deliverable
4. plans are not supposed to change mid-sprint
5. modular, iterative, and heavy on testing

# Scrum

1. Rugby/Viking term — related to *scrimmage*, *skirmish* — meant to suggest battle
2. Four components:
    a. sprint planning — stories* discussed, pokered*
    b. daily stand-up* — brief, start-of-day meeting
    c. sprint demo — show off new deliverable, etc.
    d. sprint retrospective — review, lessons learned

## Actual work consists of

1. stories — ideas for features and products
2. versions — many, normally numbered in order
3. tickets* — concrete statement of tasks, sub-tasks
4. epics — large-scale products embracing stories

# Special roles

1. project owner – makes rough drafts of stories
2. project manager – "scrum master", key person
   a. runs all meetings (and keeps them few)
   b. holds all participants accountable
   c. keeps project owner from changing things
3. scrum team – small

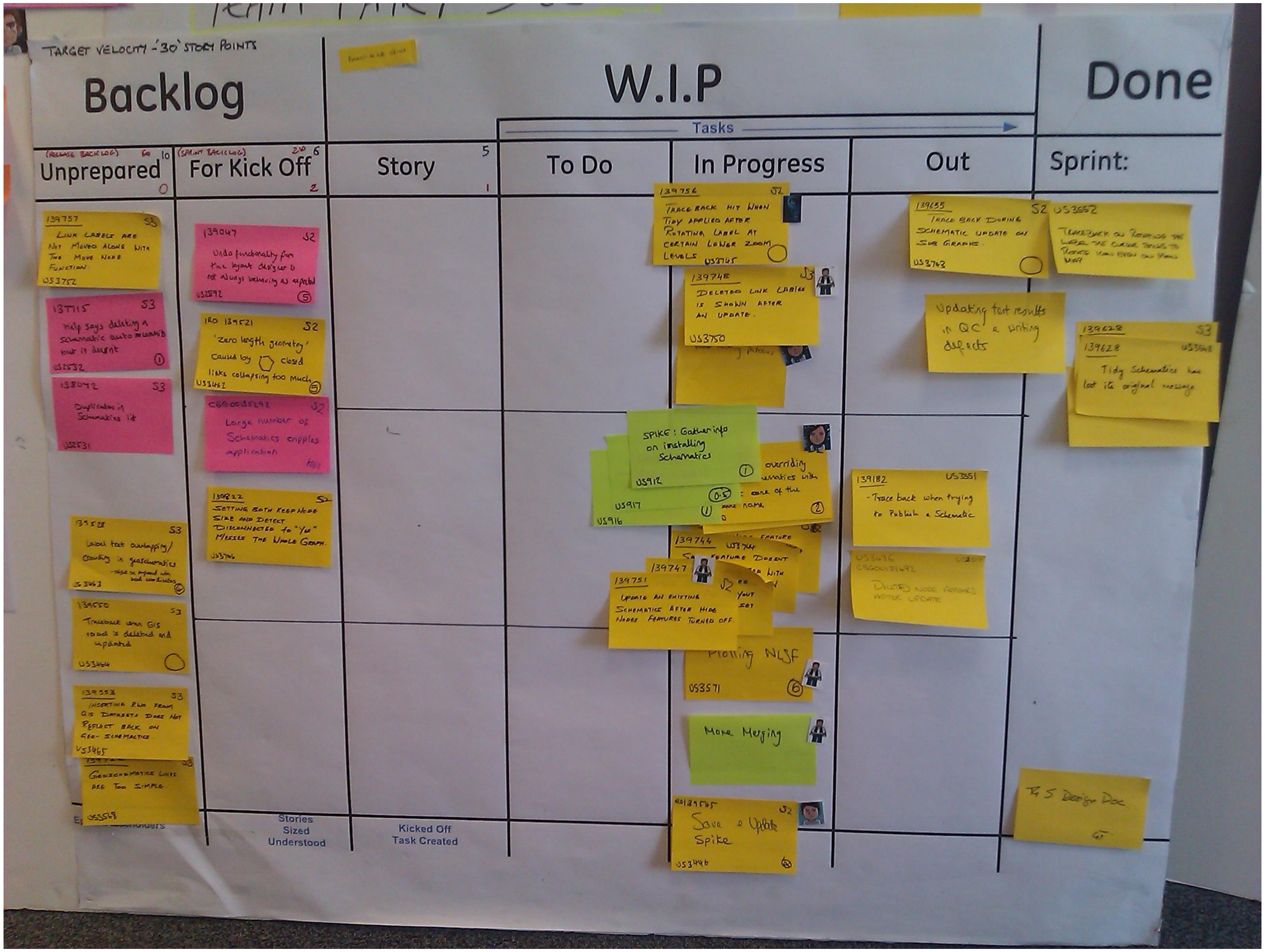
## Review of some terms

1. sprint
2. scrum
3. scrum master
4. story
5. ticket
6. stand-up
7. pokering
8. project owner
9. waterfall

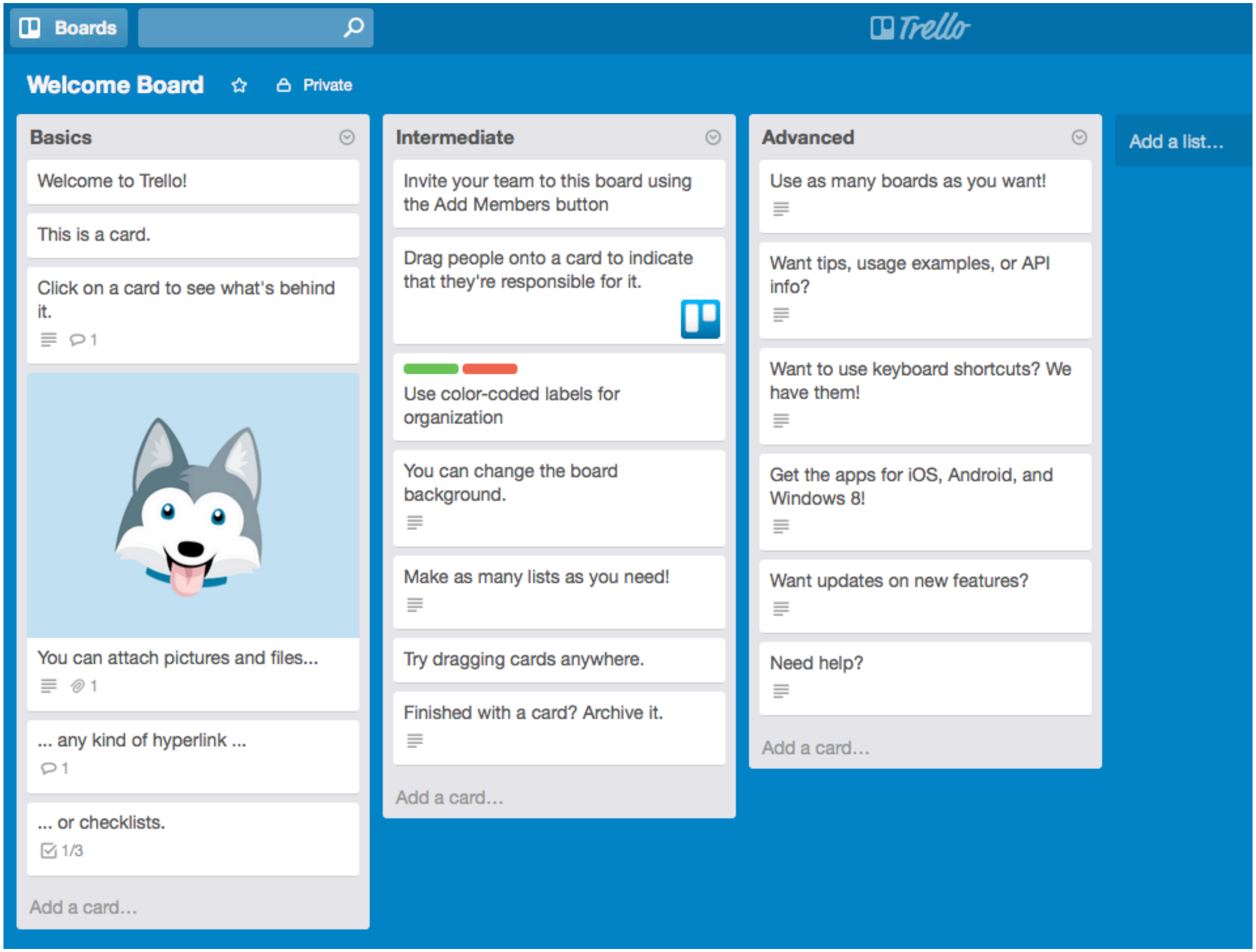
## Review of some terms

1. sprint – short, repeating development cycle
2. scrum – “skirmish”: conduct of work as battle
3. scrum master – ensures accountability
4. story – and tasks, sub-tasks, epics
5. ticket – concrete statement of tasks, sub-tasks
6. stand-up – brief, daily accountability meeting
7. pokering – assigning points (time) to tickets
8. project owner – high-level idea person
9. waterfall – not iterative/asynchronous

# Tools (1) – tickets: moveable notes on a grid


TARGET VELOCITY - '30' STORY POINTS
Backlog
W.I.P
Done
Tasks
Unprepared
For Kick Off
Story
To Do
In Progress
Out
Sprint:
Stories Sized Understood
Kicked Off Task Created
Undo functionality from the layout designer is not always behaving as expected
Help says deleting a schematic auto-reloads but it doesn't
Duplicates in Schematics list
'Zero length geometry' caused by closed links collapsing too much
Large number of Schematics cripples application
Label text overlapping/crowding in geoschematics
Traceback when GIS road is deleted and updated
SPIKE: Gather info on installing Schematics
Updating test results in QC & writing defects
Tidy Schematics has lost its original message
Trace back when trying to Publish a Schematic
More Merging
Save & Update Spike
T&S Design Doc

# Tools (2) – Trello: lightweight desktop/mobile app



**Basics**

- Welcome to Trello!
- This is a card.
- Click on a card to see what's behind it. 1
- You can attach pictures and files... 1
- ... any kind of hyperlink ... 1
- ... or checklists. 1/3

Add a card…

**Intermediate**

- Invite your team to this board using the Add Members button
- Drag people onto a card to indicate that they're responsible for it.
- Use color-coded labels for organization
- You can change the board background.
- Make as many lists as you need!
- Try dragging cards anywhere.
- Finished with a card? Archive it.

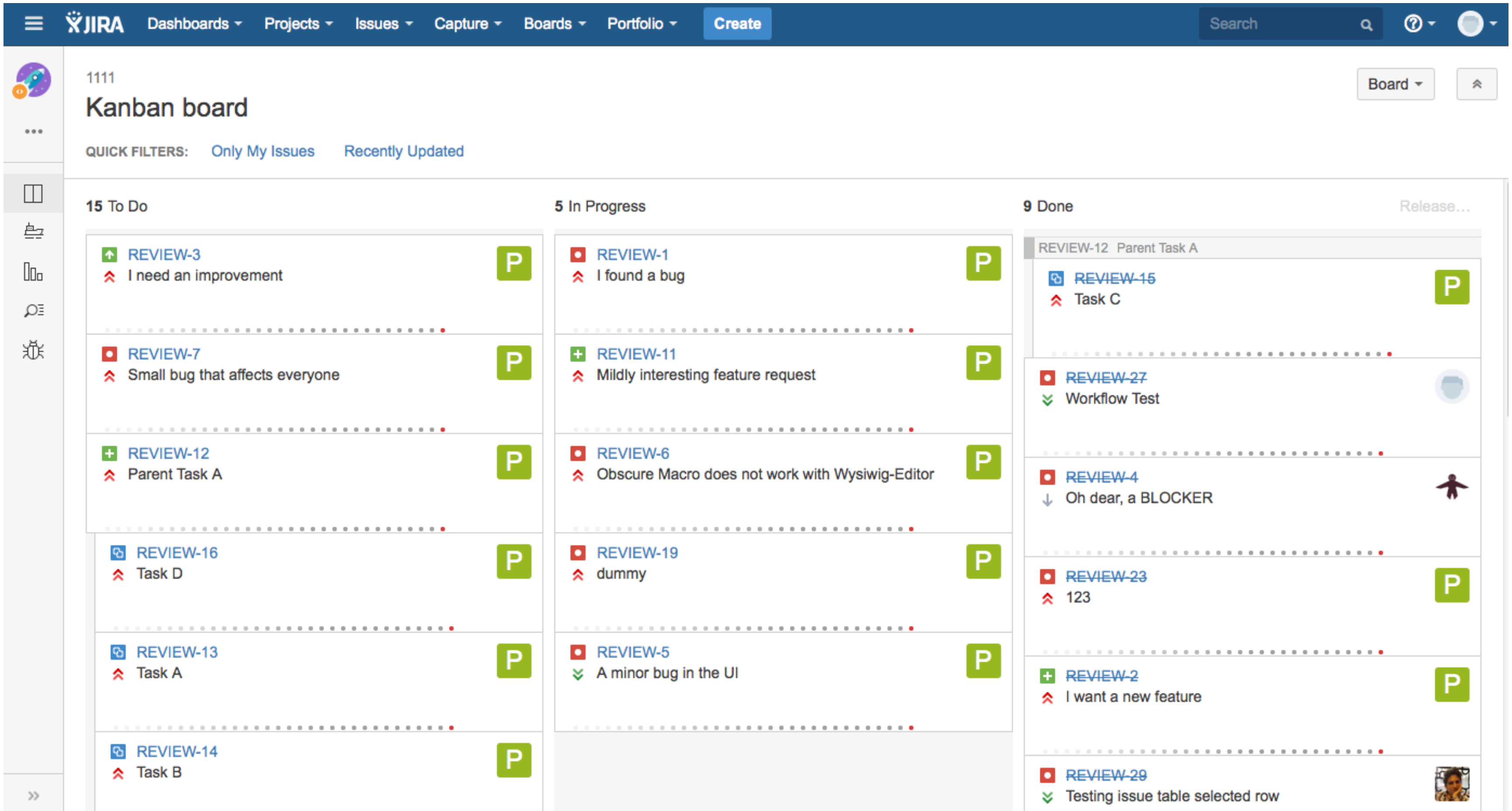
Add a card…

**Advanced**

- Use as many boards as you want!
- Want tips, usage examples, or API info?
- Want to use keyboard shortcuts? We have them!
- Get the apps for iOS, Android, and Windows 8!
- Want updates on new features?
- Need help?

Add a card…

# Tools (3) – JIRA (which I recommend)



1111

Kanban board

| 15 To Do | 5 In Progress | 9 Done |
|---|---|---|
| REVIEW-3 I need an improvement | REVIEW-1 I found a bug | REVIEW-12 Parent Task A<br>~~REVIEW-15~~ Task C |
| REVIEW-7 Small bug that affects everyone | REVIEW-11 Mildly interesting feature request | ~~REVIEW-27~~ Workflow Test |
| REVIEW-12 Parent Task A | REVIEW-6 Obscure Macro does not work with Wysiwig-Editor | ~~REVIEW-4~~ Oh dear, a BLOCKER |
| REVIEW-16 Task D | REVIEW-19 dummy | ~~REVIEW-23~~ 123 |
| REVIEW-13 Task A | REVIEW-5 A minor bug in the UI | ~~REVIEW-2~~ I want a new feature |
| REVIEW-14 Task B | | ~~REVIEW-29~~ Testing issue table selected row |

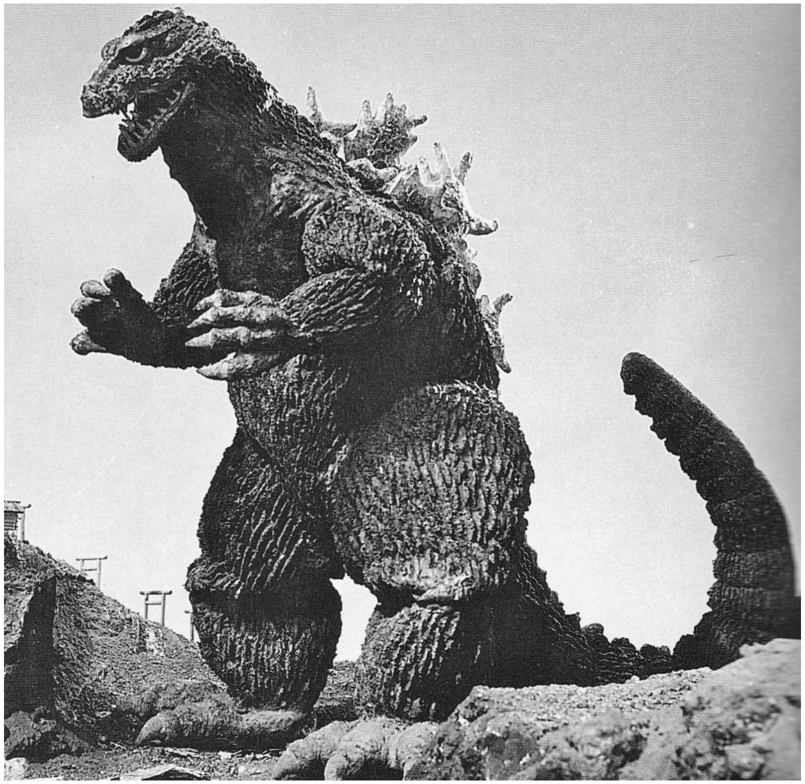
example of a JIRA board

# Tools (4) – key features (none is perfect)

1. paper/board – cheap vs. fragile, one local copy
2. Trello – free vs. limited, poor implementation
3. JIRA – some cost vs. professional, configurable

JIRA is something you can put on your résumé.

# One great thing about JIRA...



named after Godzilla – “*Gojira*“ in Japanese

originally suggested Bugzilla, Mozilla’s bug-tracker

## Links of possible interest

1. Agile Manifesto (2001). Very general — philosophy, but not a useful guide to Agile practice.
2. Websites
   1. Atlassian’s The Agile Coach site (highly recommended)
   2. Agile Alliance's “Subway Map” of practices.
   3. Amitai Schlair's Agile in 3 Minutes podcast.
   4. Peter Pilgram's Agile Software Developer Terminology for New Programmers
   5. Scaling Agile @ Spotify — how might Agile work in a very large engineering team?

劇終

end