

Agile Software Development

David Branner
Dev Bootcamp
New York

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

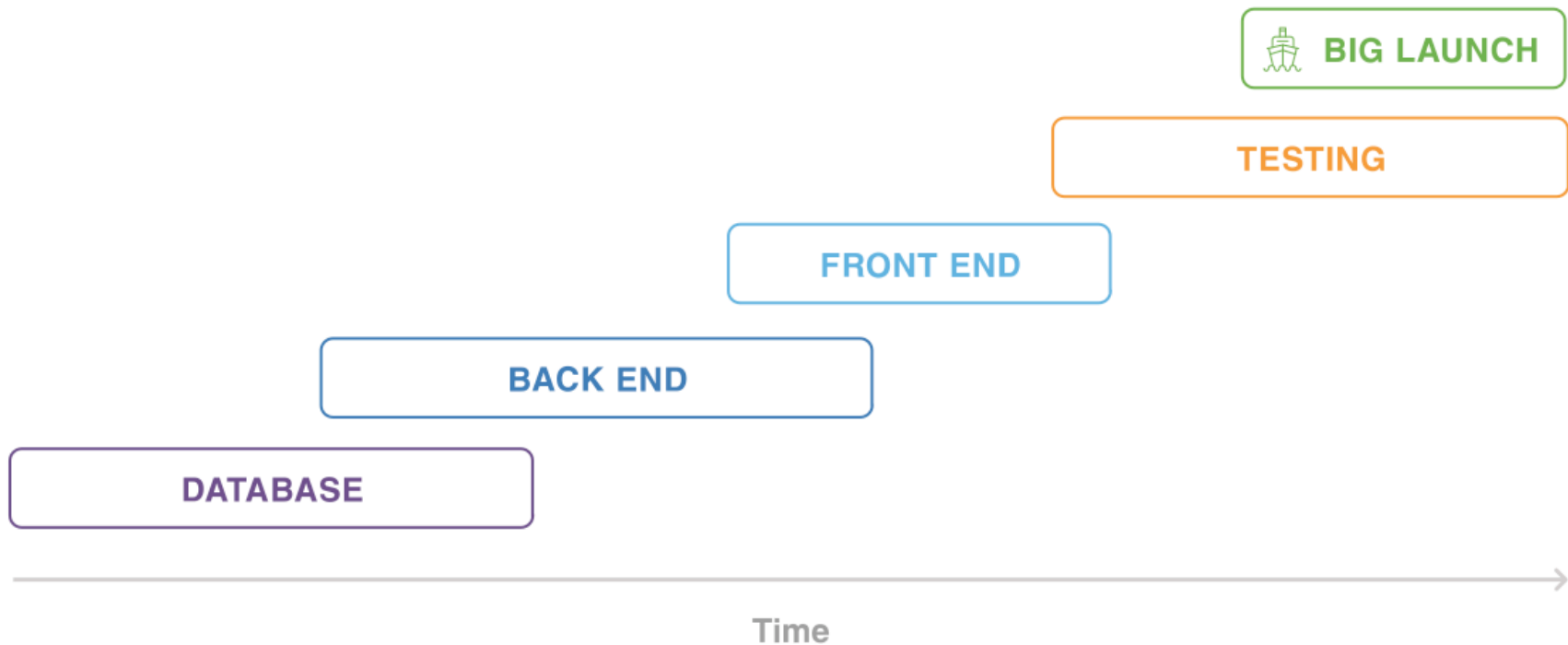
Modular: like **separation of concerns** in code and **asynchronous callbacks** in code.

Advocates contrast Agile with **“waterfall”*** **model**.

(Digression: “waterfall” model – 1)

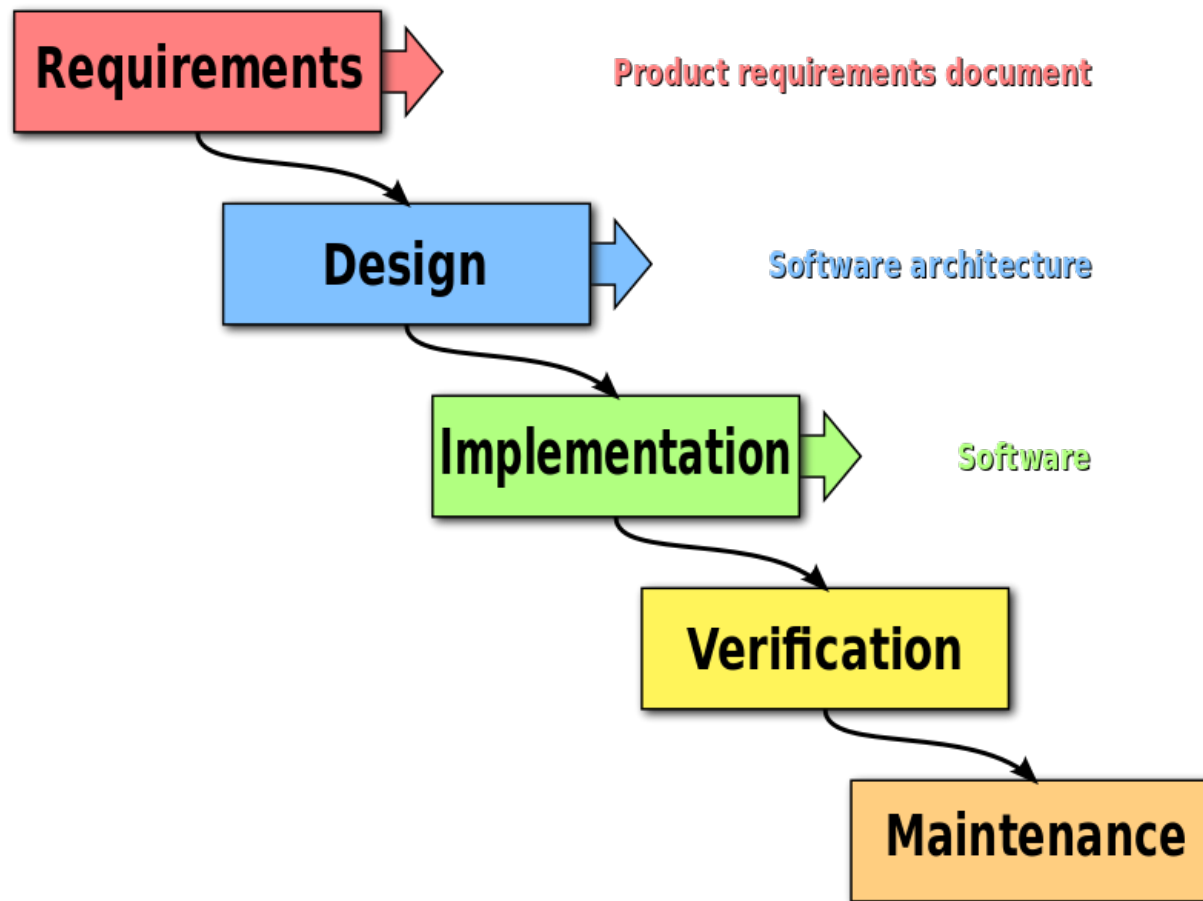
1. characterized by relatively rigid **sequence**

2. not always bad – good for material production



(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, poked*
- 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

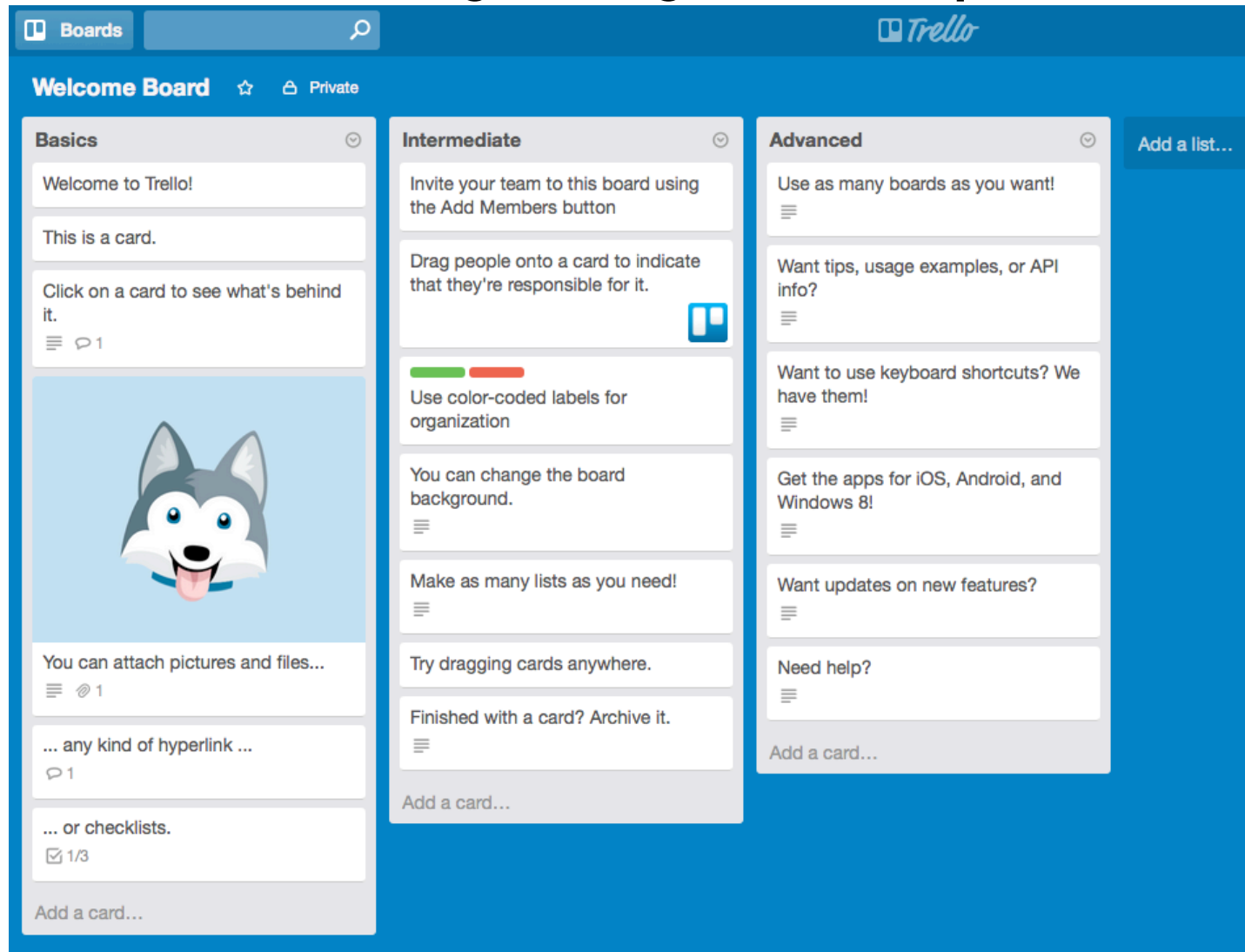
Review of some terms

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

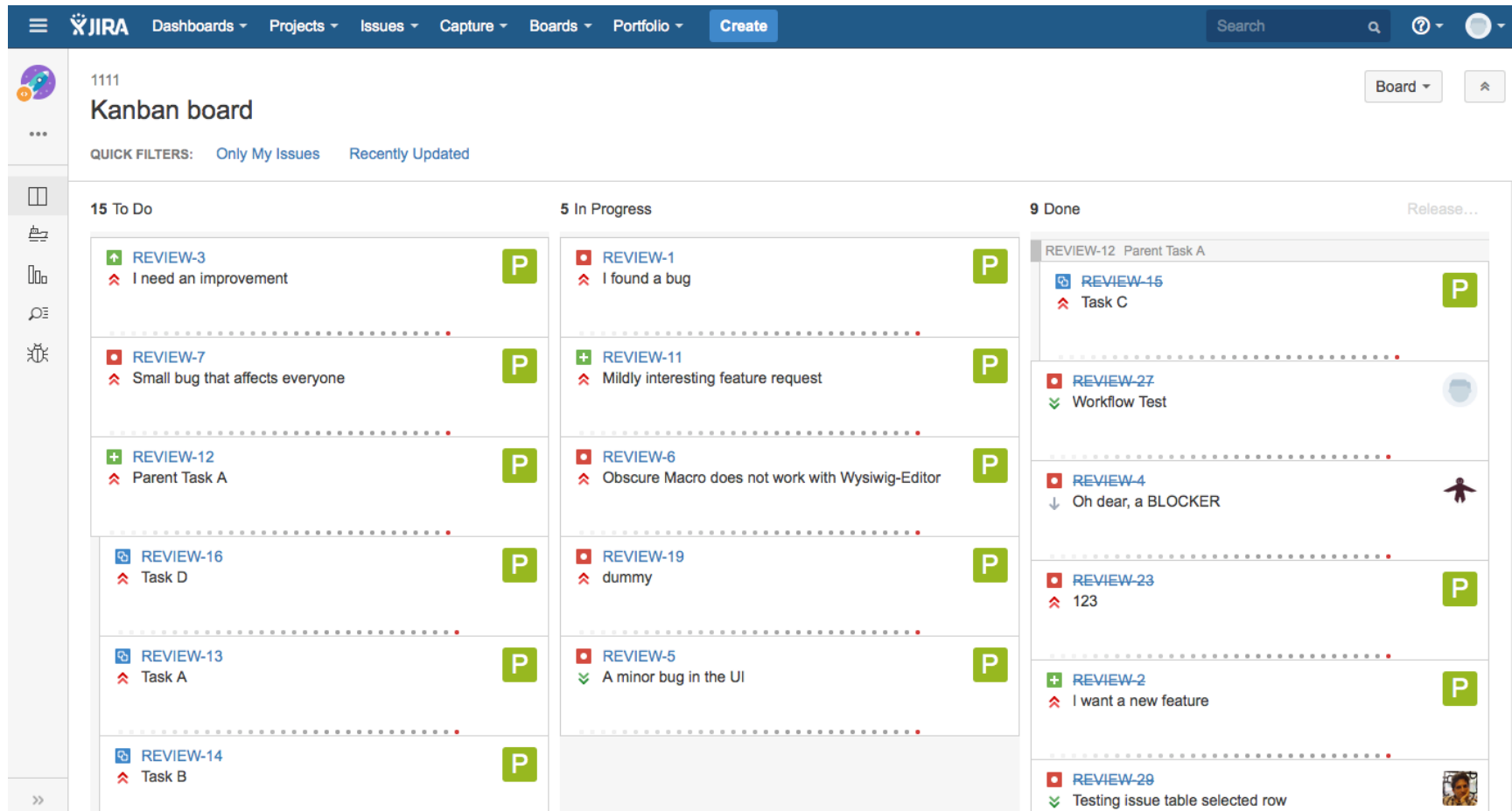
Tools (1) – tickets: moveable notes on a grid



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



Tools (3) – JIRA (which I recommend)



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

a. Atlassian's The Agile Coach site (highly recommended)

b. Agile Alliance's "Subway Map" of practices.

c. Amitai Schlair's Agile in 3 Minutes podcast.

d. Peter Pilgram's Agile Software Developer Terminology for New Programmers

e. Scaling Agile @ Spotify – how might Agile work in a very large engineering team?

劇終
end