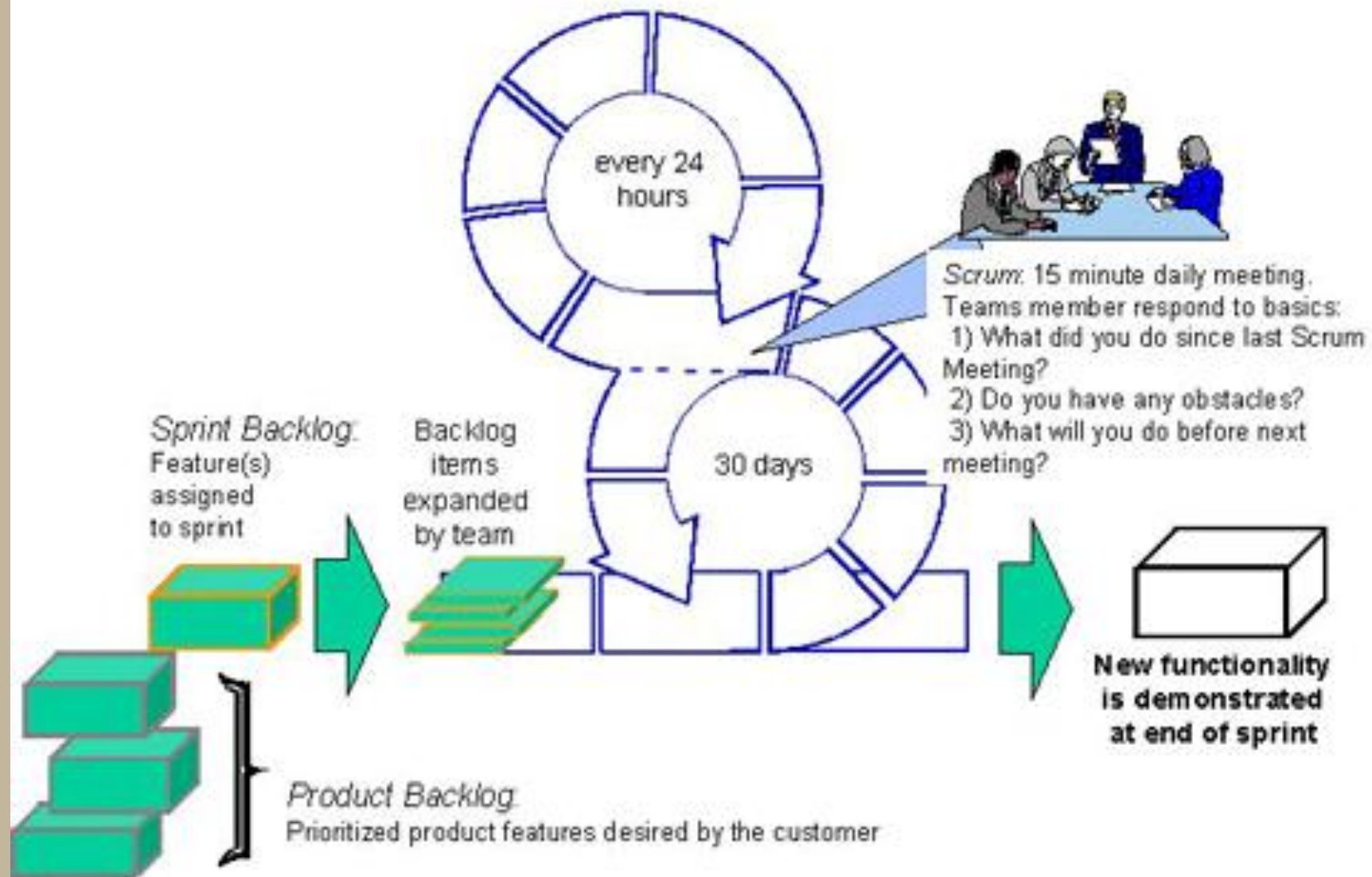# Agile / Scrum

"How-to" Guide

- *Thanks to John Lewis and Perry Reinert for contributed content*
- *Most of this content adapted from Scwaber and Beedle's Agile Software Development w/Scrum*

Scrum in a Diagram

every 24 hours

30 days

Scrum: 15 minute daily meeting. Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?

Sprint Backlog:
Feature(s) assigned to sprint

Backlog items expanded by team

New functionality is demonstrated at end of sprint

Product Backlog:
Prioritized product features desired by the customer

Who's Who

Scrum Master

- This person is responsible for *removing impediments*
- This person is *not* "the boss" or "taskmaster"
- This person is responsible for enforcing the Scrum rules
- Observes and gauges team progress
- Makes immediate decisions in the face of uncertain or incomplete information

Product Owner

- Responsible for the product
  - Should be outside the team – manager, sales, or customer
- This person owns the Product Backlog (and priorities)
- High visibility

Chickens and Pigs

- Chickens: stakeholders
- Pigs: committed

HEY PIG, I WAS THINKIN' WE SHOULD OPEN A RESTAURANT.

I DON'T KNOW. WHAT WOULD WE CALL IT?

HOW ABOUT "HAM -N- EGGS."

NO THANKS. I'D BE COMMITTED. BUT YOU'D ONLY BE INVOLVED!

By Clark & Vizdos

© 2006 implementingscrum.com

Product Backlog

- Prioritized queue of all work to be done on the product
  - *"…represents everything that anyone interested in the product or process has thought is needed or would be a good idea in the product…"*
  - As long as the product exists, the PB exists
  - It is dynamic and sorted by priority at all times
 - The Product Owner owns the Product Backlog

I need to get some stuff put on the backlog...

That shouldn't be a problem...

By Clark & Vizdos

© 2008 implementingscrum.com

Sprint Backlog

- A subset of the Product Backlog assigned to the current Sprint
  - May include tasks and subtasks identified as needed in this Sprint to finish the work.
  - The Scrum team owns the backlog
  - Scrum Master facilitates but does not own the SB
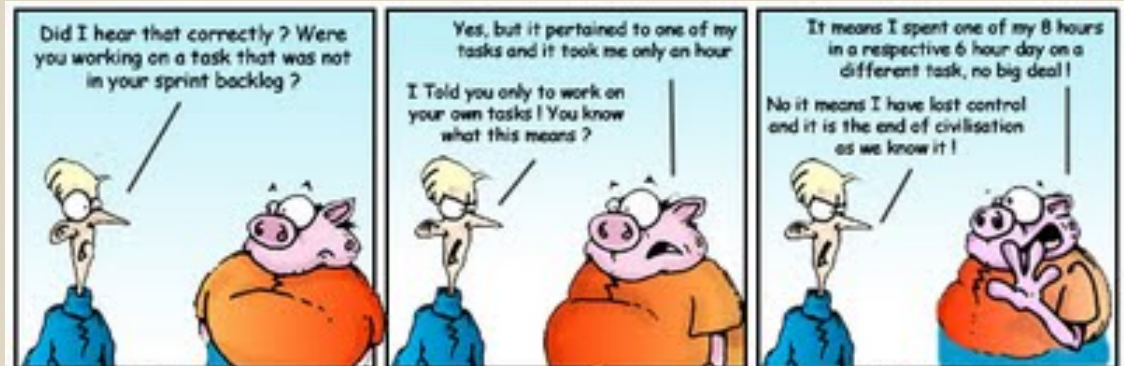
Conducting Sprints

- *Sprints* are short iterations designed to:
  - Produce working software
  - Empower the development team
  - Enable small adjustments without failing the iteration
- Sprint mini how-to:
  1. Define your Sprint Goal
  2. Conduct a Planning Session
  3. Execute the Sprint
     1. Take responsibility for a Story on the Sprint Backlog
     2. Conduct daily standups
     3. Monitor the Scrumboard, Burndown and CI/Test dashboards
     4. Large (Spike) and small (manipulate tasks) as needed
  4. Conduct a sprint *review*
  5. Conduct a *retrospective*
  6. Lather, rinse, repeat – goal is to create a *rhythm*

Sprint Steps

Sprint Goal: declarative statement of Sprint objective

– The SG can be met to different degrees

– The team may achieve SG without completing all of the SB

– Team empowered to achieve SG however they deem best

  • This is the self-organizing, autonomous philosophy in action

Sprint Planning:

– Attended by the Scrum Master, Scrum Team, and any other parties that want to see how the Sprint will likely go.

– 1st half of the meeting (~ 4 hours):

  • Scrum Team and the Scrum Master select items from Product Backlog that can be implemented in the next 30 days.

  • Scrum Team and the Scrum Master agree on a Sprint Goal:

– 2nd half of the meeting (~ 12-20 hours):

  • The Scrum Team creates the Sprint Backlog.

– Product Owner *negotiates* the Sprint Goal and Backlog to ensure the shippable increment adds business value.

Sprint Steps: Executing the Sprint

- Taking responsibility for a Release
  - The _team_ is accountable to the _Product Owner_
- Taking responsibility for a Story
  - The _Scrum Master_ is accountable for the Sprint
- Taking responsibility for a Task
  - Developers are _accountable_ to the _team_
- **Suggested Machinery**
  - Developers should only be working on a small number of tasks at a time (some believe small = 1)
  - Minimize the number of devs working on a story
  - Change the tasks as you see fit
 - Done means <u>DONE</u>, all through way through test

Dev teams are empowered – to estimate, determine tradeoffs, and decide on processes. They accept responsibility for a deliverable and are accountable but get to decide how to do it!

ASU

Two types of Stakeholders

- *Business people*: Responsible for Scope, Priority, and the Composition of specific releases: whether a release meets a threshold criteria
- *Development people*: Responsible for Estimates
  - Consequences: tradeoffs, or effects of business decisions on technical feasibility
  - Process: What specific tasks to do
  - Detailed Scheduling: per release, which stories first?
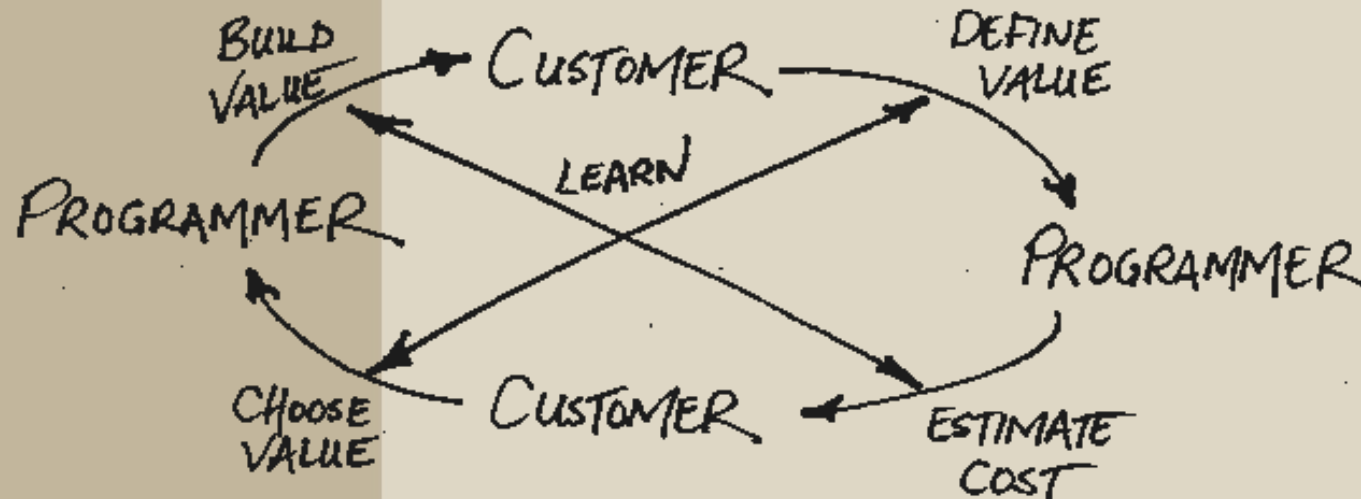
"Lather, Rinse, Repeat" - Every Sprint

ASU

Daily Standup or "Scrum"

– Attended by Chickens and Pigs

– Each team member answers 3 questions

- "Since last Scrum I…"
- "Before next Scrum I will…"
- "My impediments are…"

– Scrum Master runs it

- Documents impediments
- Must later help remove them
- Makes all "now" decisions

– The questions should be answered with respect to specific User Stories and/or Tasks in the Sprint Backlog, and reference specific work items (commits) in the SCM

- "I did some stuff" doesn't cut it – be measurable!!!

– https://www.youtube.com/watch?v=pWS8Mg-JWSg

The Scrumboard: An *Information Radiator* (Cockburn)

- Maps out the stories and tasks for all to see
- Variants exist, but work goes through at least 3 states
  - To-do – no one has picked it up on the team yet
  - In-process – assigned to a team member (a "pig")
  - Done – assigned member completed it
  - I also like "Verify" or "To-test" but your mileage may vary
- When all tasks are done the story should be done
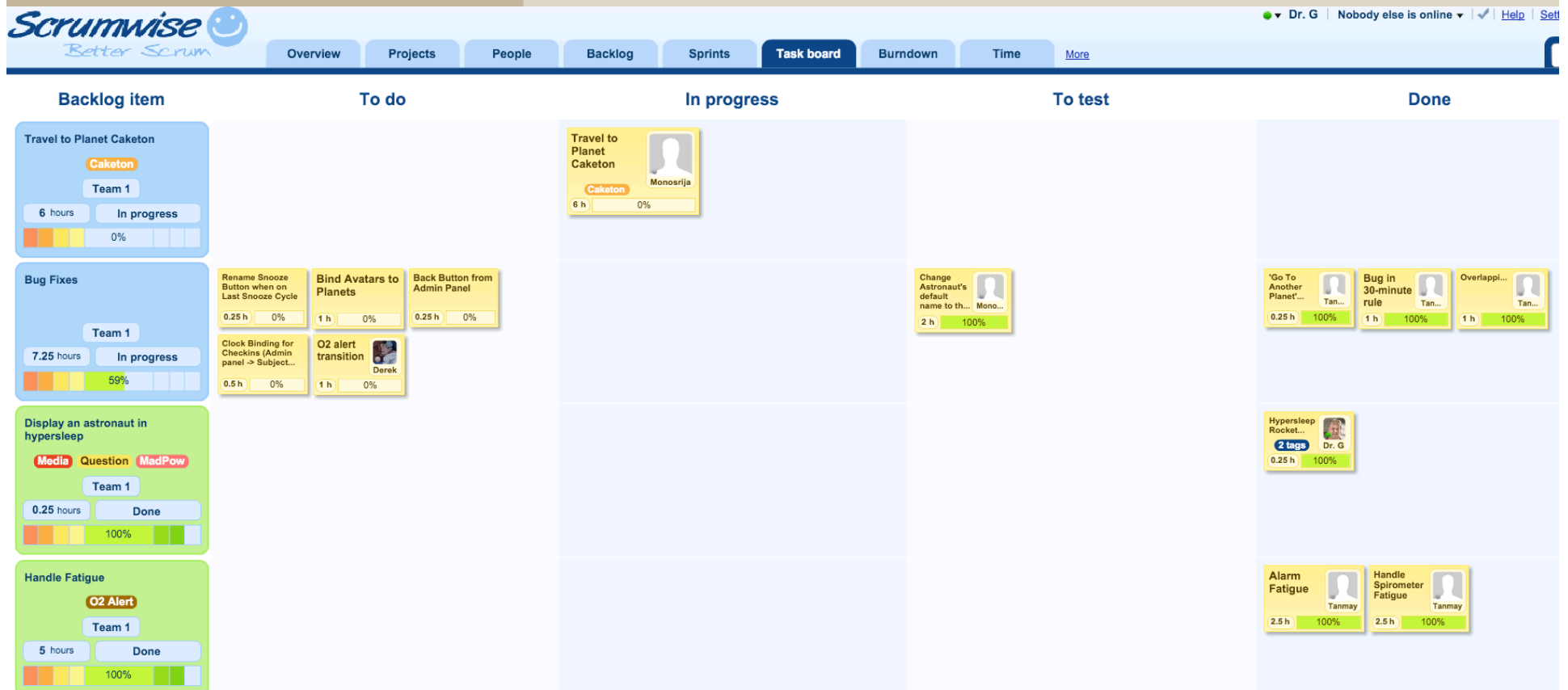- When all stories are done the Sprint Goal should be met
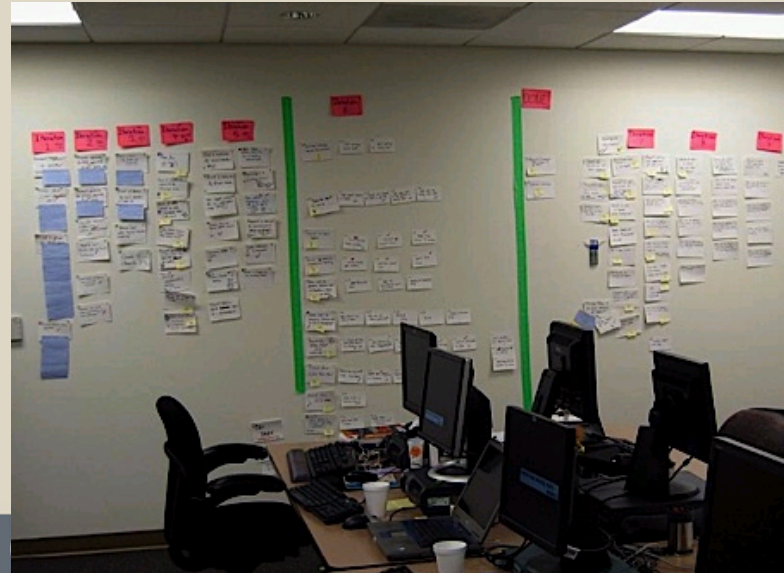
Example e-Scrumboard

- Fair number of online Scrumboard tools exist now to support distributed teams. Example in Scrumwise:

POLYTECHNIC CAMPUS

Sprint Steps: Executing the Sprint

Steering – embracing change

– Executing a Sprint has been compared to driving a wobbly vehicle on a slick road; you always want to head a certain direction but you make constant adjustments to get there!

– In Scrum, the Daily Standups and Information Radiators give you constant data to which you _react_

– Team may change its tasks, but not stories or Sprint Goal!

Spike – dealing with blockers

– The entire team focuses its collective energy for issues like:

1. The need to do a "deep dive" into a design/technology question with heavy resources to gain information

   • You "spike" because you cannot afford to get the decision wrong!

   • Represented as a "0-value" story on the Scrumboard

2. When there is an obstacle facing the team that is interrupting the team's "velocity" – all resources stops and focus on helping the developer overcome this obstacle before returning to their work.

- When a Sprint completes, the Sprint Team gives a demo of working software to the Product Owner
  - <=2 hours of preparation allowed.
  - This is a purely functioning demo; no use of PowerPoint.
  - All interested in seeing the results of the Sprint attend.
    - A decision is made as to whether the Sprint Goal has been met
  - Purpose is to solicit feedback. Observations and comments will commonly become items on the Product Backlog.

It's my project...and It's not done until I say I'm done.

BZZZZT!

I'd say he's done. Let's move on...

By Clark & Vizdos

© 2006 implementingscrum.com

- A *postmortem* or *project review* is an accepted software engineering process practice
  - Conducted at the end of projects
  - Focuses on process improvement across the organization
  - And yes, a chance to vent.

- *Sprint Retrospectives* are not as comprehensive
  - Addresses what is going in with the team today
  - Is data-driven; the team collects quantitative and qualitative measures of how it is doing
  - Suggests improvements that can be implemented today
  - Often recommendations are not earth-shattering re-definition of processes or tools, but merely small changes (sound familiar?) or reminders on what needs to be done.

Questions?