

BETTER(SMALL) SOFTWARE TEAMS

Michael A. Heroux

types.Operator):
 'X_mIRROR to the selected object.mirror_x'
 'mirror X'

Outline

Small Team Characteristics

- Models
- Challenges.

Small Team Agile workflows

- Intro to terminology and approaches
- Overview of Kanban
- Free tools: Trello, GitHub.

Small team interaction model

Team composition:

Senior staff, faculty:

- *Stable presence, in charge of science questions, experiments.*
- *Know the conceptual models well.*
- *Spend less time writing code, fuzzy on details.*

Junior staff, students:

- *Transient, dual focus (science results, next position).*
- *Staged experience: New, experienced, departing.*
- *Learning conceptual models.*
- *Write most code, know details.*

Large team challenges

Composed of small teams (and all the challenges).

Additional interaction challenges.

Policies, regularly cultural exchanges important.

Small team challenges

Ramping up new junior members

Background

Conceptual models

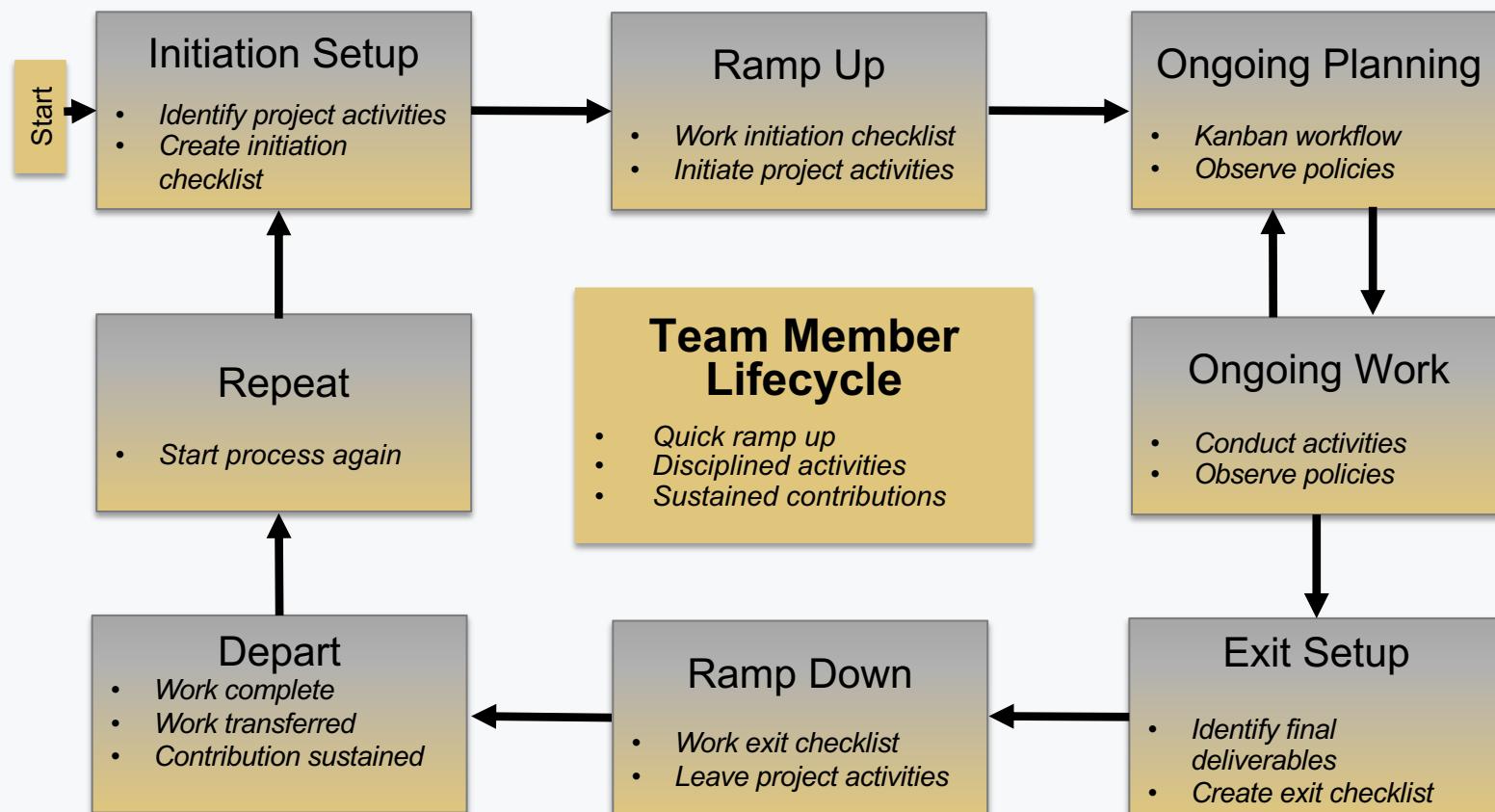
Software practices, processes, tools

Preparing for departure of experienced juniors

Doing today those things needed for retaining work value

Managing dual focus

Team Member Lifecycle



Checklists & Policies

Team Member Phase		
New Team Member	Steady Contributor	Departing Member
Checklist	Policies	Checklist

Use checklists to

On ramp new team members

Ramp up a new project

Ramp down finished project, prepare for departing team member

Use policies

To maintain and improve quality

Assure sustainability in preparation for lottery events

Managing issues: Fundamental software process

Issue: A bug report or feature request

Approaches

Short-term memory, office notepad

ToDo.txt on computer desktop (1 person)

Issues.txt in repository root (small co-located team)

...

Web-based tool + Kanban (distributed, larger team)

Web-based tool + Scrum (full-time development team)



Informal, less training

Formal, more training

Kanban Principles

Limit number of “In Progress” tasks

Productivity Improvement

Optimize “flexibility vs swap overhead” balance. No overcommitting

Productivity weakness exposed as bottleneck. Team must identify,fix bottlenecks

Effective in R&D setting. Avoids deadline approach. Deadlines can be added.

Provides a board for viewing and managing issues

Can be applied to any existing software project *immediately!*

Task: Have Eureka moment by Tuesday.

Scrum

Basic Kanban

Backlog	Ready	In Progress	Done
<ul style="list-style-type: none">• Any task idea• Trim occasionally• Source for other columns	<ul style="list-style-type: none">• Task + description of how to do it.• Could be pulled when slot opens.• Typically comes from backlog.	<ul style="list-style-type: none">• Task you are working on <i>right now</i>.• The only kanban rule: Can have only so many “In Progress” tasks.• Limit is based on experience, calibration.• Key: Work is <i>pulled</i>. You are in charge!	<ul style="list-style-type: none">• Completed tasks.• Record of your life activities.• Rate of completion is your “velocity”.

Notes

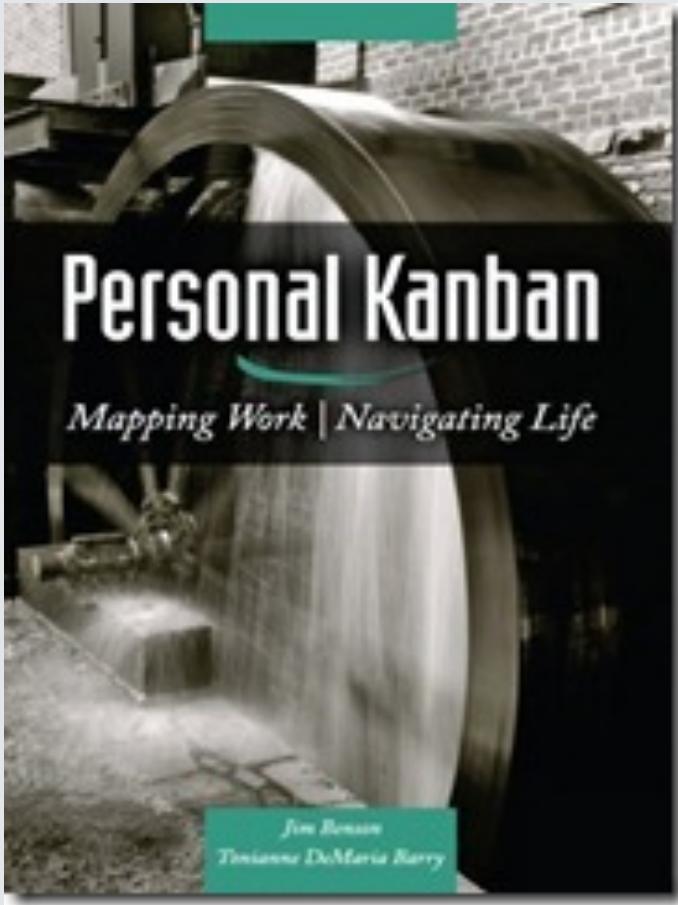
Ready column not strictly required, sometimes called “Selected for development”

Other common column: In Review

Can be creative with columns

Waiting on Supervisor Confirmation

Tasks I won’t do



Personal Kanban

Personal Kanban: Kanban applied to one person.

Apply Kanban principles to your life.

Fully adaptable.

Personal Kanban: Commercial book/website.

Useful, but not necessary.

<http://www.personalkanban.com>

Kanban tools

Wall, whiteboard, blackboard: Basic approach

Software, cloud-based

Trello, JIRA, GitHub Issues

Many more

I use Trello (browser, iPhone, iPad)

Can add, view, update, anytime, anywhere

Question: How many tasks?

Personal question

Approach: Start with today's in-progress tasks. See how it goes

Use a freeway traffic analogy:

Does traffic flow best when fully packed? No.

Same thing with your effectiveness.

Spend time consulting board regularly

Brings focus

Enables reflection, retrospection

Use slack time effectively

When you get out of the habit, start up again

Importance of “In Progress” concept for you

Junior community members

Less control over task.

Given by supervisor.

In Progress column: Protects you

If asked to take on another task, respond:

- *Is this important enough to become less efficient?*
- *Sometimes it is*

Key Team Management Elements

Checklists:

Initiation, Transition, Exit

Policies:

How team conducts its work

Issue tracking system:

All work tracked, visible to team

Milestones: Aggregate related issues.

Kanban board

Regular meetings, updates

PERSONAL EXPECTATIONS

Calling out the best in team members

Final Thoughts: Commitment to Quality

Canadian engineers' oath (taken from Rudyard Kipling):



*My Time I will not refuse;
my Thought I will not grudge;
my Care I will not deny
toward the honour, use,
stability and perfection of
any works to which I may be
called to set my hand.*

A Few Concrete Recommendations

Show me the person making the most commits on an undisciplined software project and I will show you the person who is injecting the most technical debt.

GitHub stats: Easy to find who made the most commits

Some people: Pride in their high ranking

Instead, be the person who ranks high in these ways:

Writes up requirements, analysis and design, even if simple

Writes good GitHub issues, tracks their progress to completion

Comments on, tests and accepts pull requests

Provide good wiki, gh-pages content, responses to user issues

<https://github.com/trilinos/Trilinos/wiki/Productivity---Initiative>

(Personal) Productivity++ Initiative

Ask: Is My Work _____ ?

Productivity++

- ✓ Traceable
- ✓ In Progress
- ✓ Sustainable
- ✓ Improved

