Agile & Efficient Computing: Course Review

MICHAEL A. HEROUX

Course Goals

Improve software skills

Making the right product:

- Understanding and delivering what the stakeholders want
- Being effective

Making the product right:

- Using good processes and tools
- Being efficient

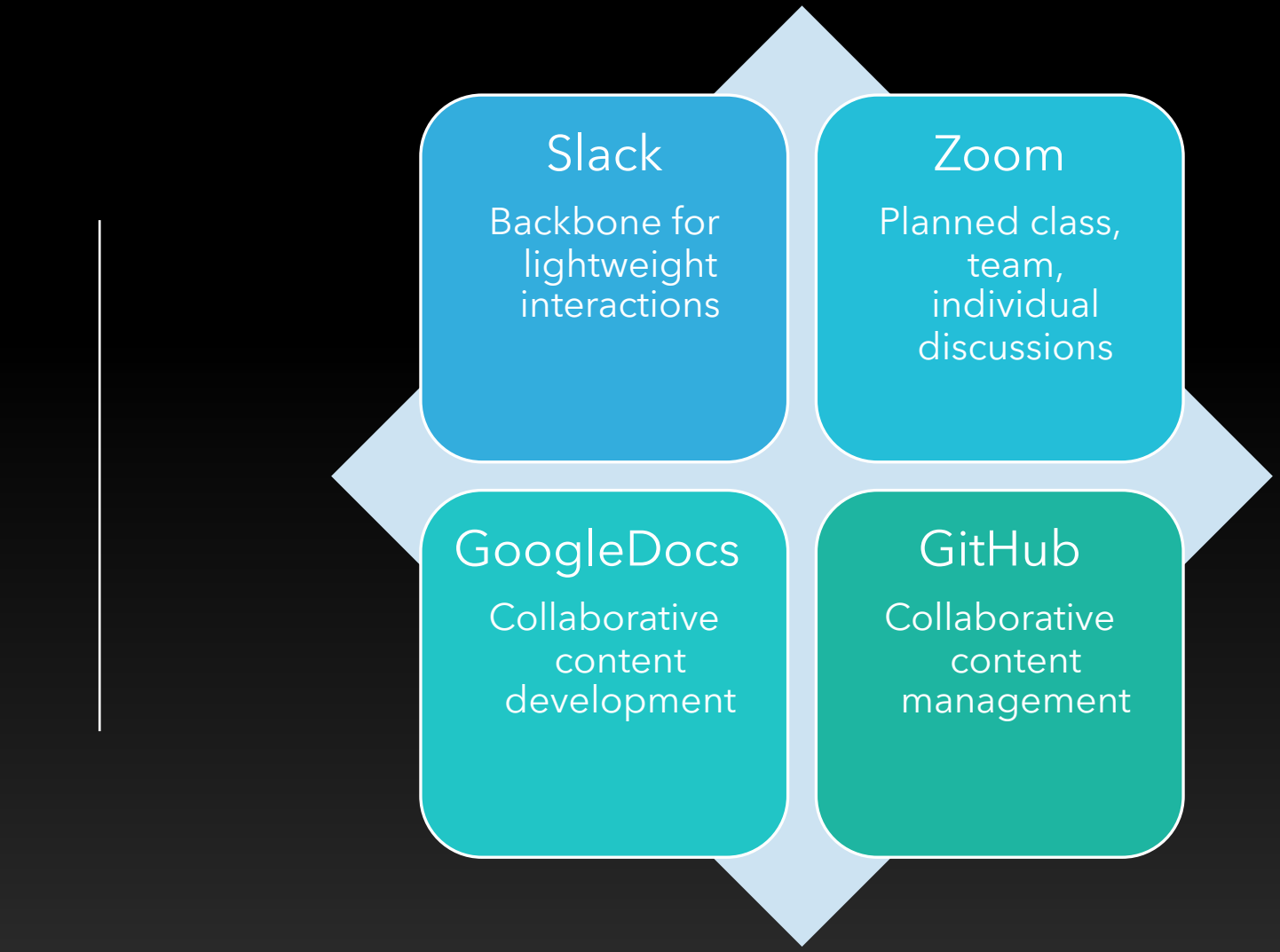
Use computing for insight

Use data, algorithms in simulations

- Random sampling for gnarly problems
- Past results for future prediction

Estimate and simulate for decision support

Communication tools



Slack: Creating persistent presence

Enables a virtual team room experience

Have it on your phone, desktop, tablet, Apple Watch

Monitor it regularly

Expectations

Keep course goals in mind

- Effective and efficient software product development
- Insight using computing

Be transparent

- Communicate to me and your team about opportunities, challenges
- Communicate early, when issues are just emerging and easier to address

Commit

- Stay focused
- Adapt as needed

Software Projects

Pi Estimator
- Individual project to jumpstart our efforts
- Java + random numbers

Solitaire Simulator
- Team project
- Use Monte Carlo techniques to estimate winning percentage
- Improve software design skills
- Use a new language (Python or C++)

March Madness
- Team project
- Use data (and basic database concepts) and analysis to predict team rankings

Major Software Topics

Building Effective Teams:	• Team policies • Checklists
Agile Requirements Techniques:	• User Stories • Epic-Story-Task hierarchies
Design Approaches:	• Diagrams • Cohesion and Couple
Assessment Techniques	• Review strategies

Project Management

Project Tuning Knobs:

- Cost - $
- Scope – Work to be done
- Schedule – Timeline

Kanban:

- Backlog, In Progress, Done, and more
- Alternative to Scrum
- Kanban-and vs Scrum-but

User Stories: “As a < >, I want < >, so that < >.”

- **User Stories:** Lightweight method for defining, refining, prioritizing requirements.
- **Resource:** Many sources. We list just a few:
  - User Stories – Atlassian: Makers of Confluence, Jira.
  - User Stories: An Agile Introduction: Agile Modeling
  - User Stories: Commercial site (Mountain Goat Software), but good basics.
  - How to write good user stories: A YouTube video from CA Technologies that provides some tips for useful stories.
  - Getting Started with Agile : Epics, Features, and User Stories - YouTube: A nice overview of agile requirements.
  - “As a, I want, So that” Considered Harmful: Does not dismiss value of user stories, reminds the reader that the format is not magic, and that variations can be useful for encoding requirements more effectively. More generally, Crisp's Blog (note: many articles are in Swedish) is a good resource for topics in modern software development.
- Replacing the User Story with the Job Story: “When < >, I want to < >, so I can < >.”

Stories Useful in Many Settings

Tool selection

Select team IDE - Cloud9 AWS – GoogleDocs of IDEs.

Team policy

Behaviors, practices expected from my team.

Group reorg

What each team member needs – Guided ECP ST reorg.

Regrouping

Remembering priorities as the project progresses

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

Basic Kanban

Backlog	Ready	In Progress	Done
• Any task idea • Trim occasionally • Source for other columns	• Task + description of how to do it. • Could be pulled when slot opens. • Typically comes from backlog.	• Task you are working on *right now.* • **The only kanban rule: Can have only so many "In Progress" tasks.** • Limit is based on experience, calibration. • **Key: Work is *pulled*. You are in charge!**	• 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

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*

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

Basic Agile (Software) Process Stages

RADID

Requirements	Create User Stories
Analysis	Refine stories
Design	Describe in words, pictures
Implementation	Create software
Delivery	Product to users

Testing?

Testing is not a stage

- Integral to all stages

Test analysis

- Does it address requirements?

Test design

- Does it address analysis outcomes?

Test software

- Does it address design goals?

Test delivery

- Will the software work for the user?

Are Stages Really Stages?

- Yes, but conducted incrementally per feature
- Multiple product features can be “in flight” at once
- Each feature can be in a different stage
- Each stage of each feature tracked and managed as task
- Testing of each stage is also a task

Basic Agile (Software) Project Activities

PETA