Agile @ Wikimedia

Tomasz & Arthur

Today

- What is ‘agile’?
- Team: E2/Flow
- Team: Mobile Web
- Roles and responsibilities
- … the next month

What is ‘agile’?

What is ‘agile’?

… a group of software development methods based on **iterative and incremental development**, where requirements and solutions evolve through **collaboration between self-organizing, cross-functional teams**. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change.

-- http://en.wikipedia.org/wiki/Agile_software_development

Agile manifesto

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Agile principles (paraphrased)

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Working software is the principal measure of progress
5. Sustainable development, able to maintain a constant pace
6. Close, daily cooperation between business people and developers
7. Face-to-face conversation is the best form of communication (co-location)*
8. Projects are built around motivated individuals, who should be trusted
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular reflection and adaptation to evolving needs

Agile vs agile*

Or… What we mean when we say ‘agile’

*http://apsblog.burtongroup.com/2008/09/---font-definit.html

Dogmatic adherence to process

Image cc by-nc-sa, Leo Reynolds
http://www.flickr.com/photos/lwr/104105346/in/set-72157594159047168

A state of mind

Image cc by-nc-sa, Leo Reynolds
http://www.flickr.com/photos/lwr/4878024477/in/set-72157594159047168

21
8
GARMIN

Scrum

Team: **E2/Flow**

- What’s working?
- What’s not?
- What confuses us?

Etherpad: ***http://bit.ly/16hUyvW***

Break!

Team: **Mobile Web**

Success stories of the Mobile Web Team

Problem: What should I work work on, and how do I know when I’m done?

- Frequent context switching
- Little overlap with team members
- Not knowing if your done
- Needing to ask manager(s) what to do next - sometimes receiving conflicting information

Solution: Sprints, User Stories, Acceptance Criteria

- **Sprint**: A fixed period of time in which to build features from user stories, in priority order.

- **User Story**: One or two sentences written in everyday language of an end user that captures something they want to do with your software, and why. E.g., “As a registered user, I want to be able to log in to the website, so that I can use the site’s advanced features.”

- **Acceptance criteria**: Definitions of the expected outcome of a given task. E.g., “When I enter an incorrect password at login and press ‘submit’, I am returned to the login page and displayed an error.”

Outcomes:
Predictability and Autonomy

Problem: Who defines the work and priorities?

- Numerous stakeholders asking for work
- Conflicting opinions about the relative importance of tasks
- No single voice
- Endless debates

Solution: Product Owner

- Product owner: Responsible for the overall ‘product’, and has final authority over the priorities of work to be done.

Outcome:

Forward motion

Problem: No way to predict how much work we could accomplish

- Difficult to make accurate commitments to stakeholders or other teams
- Easy to promise too much
- Working late and on weekends
- Difficult to plan ahead

Solution: Estimation and measurement

- **Relative Estimation**: Applying a point value to a user story to indicate the amount of effort it would take to accomplish, relative to a known baseline.
- **Measurement:** Analyzing how many story points the team has completed in previous iterations gives a good indication of many story points they will complete in future iterations.

Outcomes:

Predictability, Deliverability, and Increased Collaboration

Problem: No process for team improvement

- Unclear paths for conflict resolution
- Team-members complain to the manager
- Manager responsible for implementing change on the team

Solution: Retrospectives

- **Retrospective:** Timeboxed meeting for the team to discuss what has been working well and what hasn’t. Pain points are prioritized and assigned owners responsible for implementing improvements.

Outcomes:

Team ownership, self sufficiency, and healthier relationships

Problem: Difficulty communicating across team peer groups

- High tension conversations
- Group infighting
- Decisions made not always well communicated across different peer groups

Solution: Rule of 3

- Always involve at least one member from Design, Product, and Engineering in discussions

Outcome:

Balanced and High Quality Product

Problem: Isolation

- Engineers working in silos, sometimes unaware of components other engineers are working on
- Frustration stemming from aforementioned problems with no clear path for resolution
- Remote employees feeling disconnected

Solution: Rituals, norms, video chat

- **Rituals**: Routine ~~sacrifices~~ timeboxed meetings to connect, plan, and evaluate as a team. E.g. daily standup
- **Norms**: Collaboratively agreeing on guidelines around communication, how we engage our work, and how we engage each other. E.g. “If it didn’t happen on the mailing list, it never happened.”
- **Video chat**: Employed in every meeting, this helps minimize the perceived distance between team members.

Outcome:

We are all remoties

Roles and Responsibilities

Roles overview

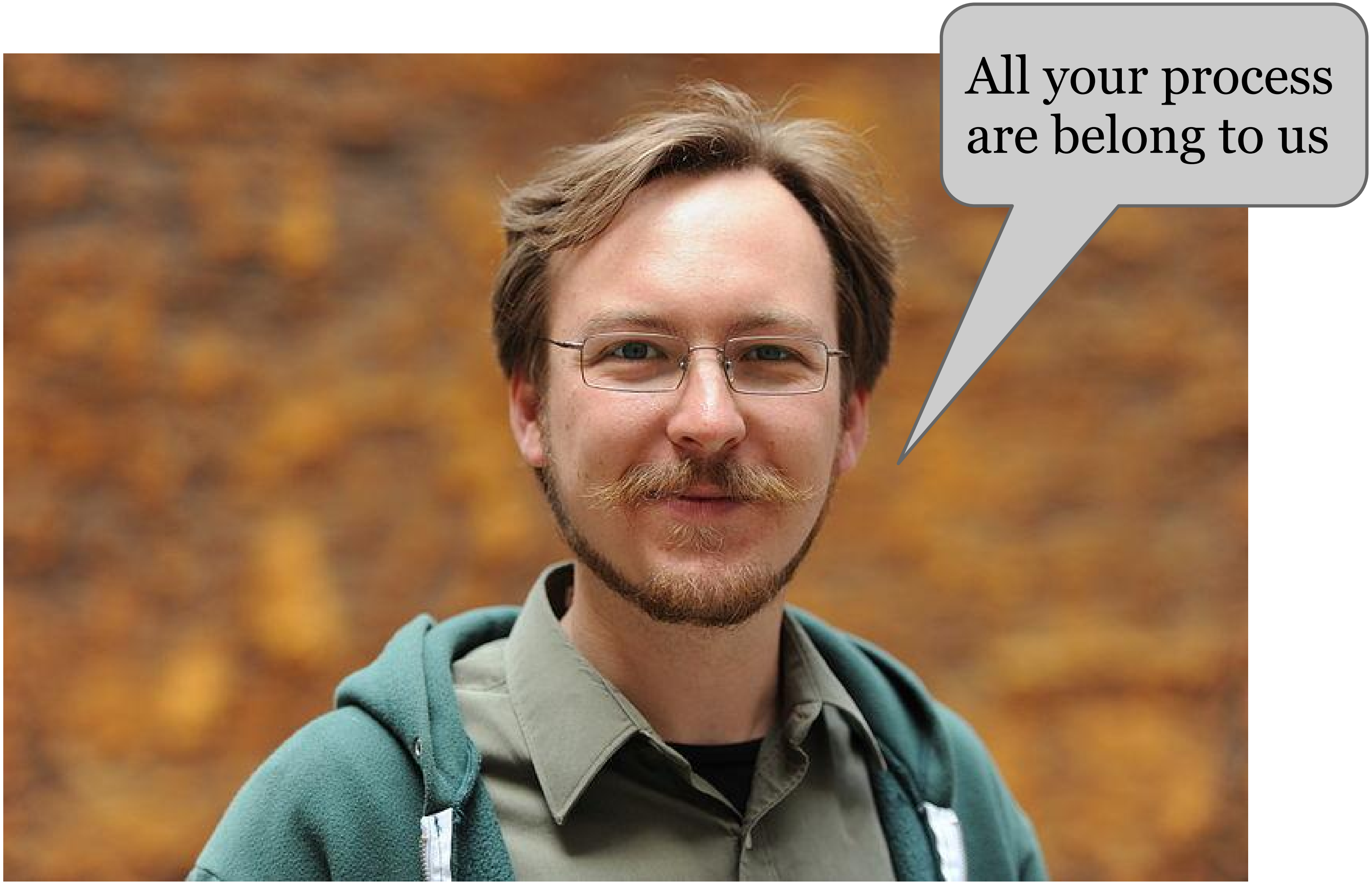
- Product owner
- Scrummaster
- Developer
- Tech lead
- UX/UI designer
- QA analyst
- Business analysts

Product Owner

Product owner

- Provides vision for the product or service being developed
- The **single point of escalation for contending priorities** among stakeholders
- Manages the product road map
- **Makes final decisions about trade-offs** when desired functionality, or scope, exceeds the capacity of the team
- Defines the target constituent for iterations, releases, and the overall product
- Makes final decision about whether or not work done on stories is complete ('acceptance').
- Determines what features the team should work on in order to achieve our user, community, and Foundation goals
- Makes sure that products have a coherent, compelling user experience

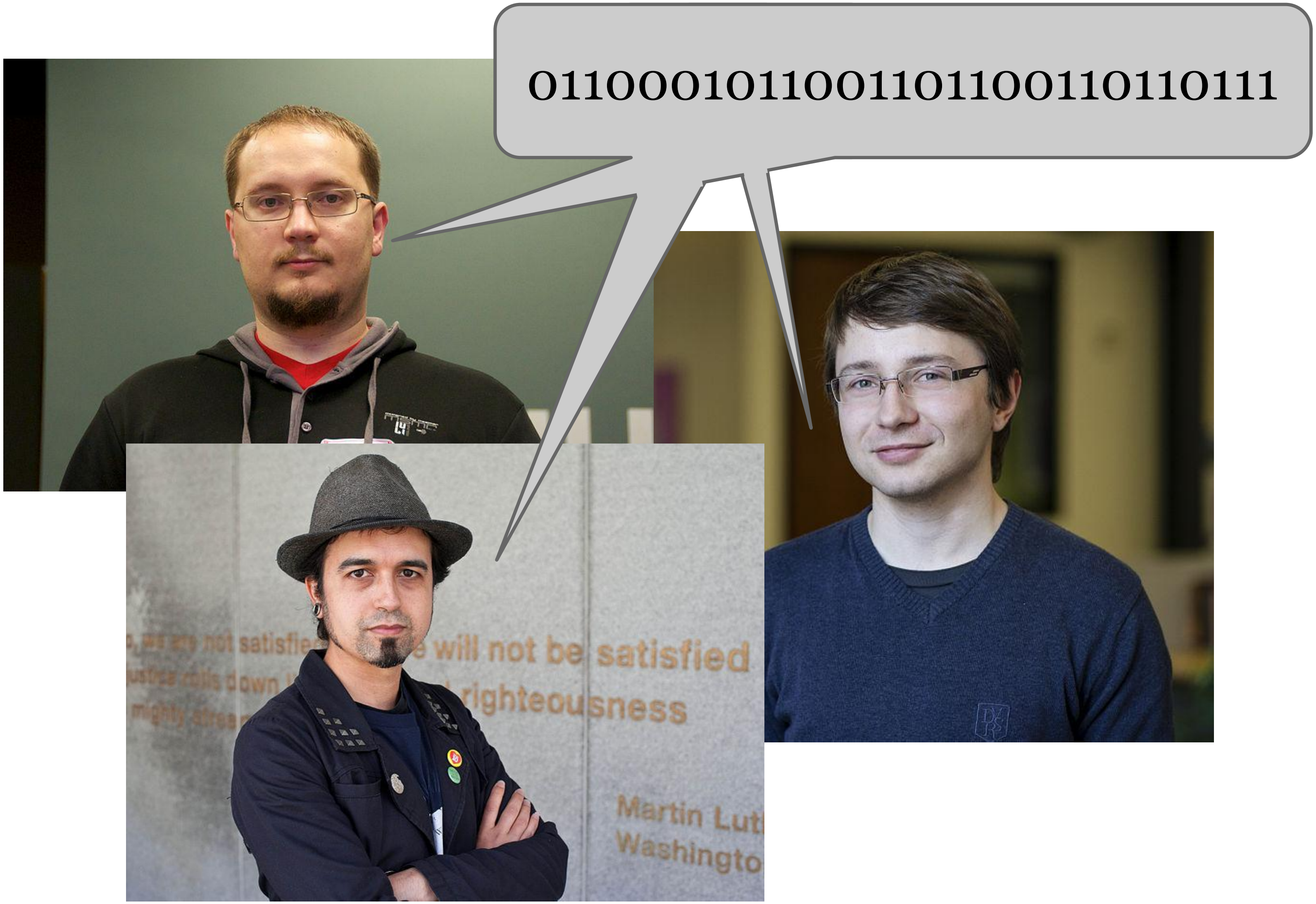
Scrummaster



Scrummaster

- Evolves team/agile capacity
- Removes impediments to development
- Leads/coordinates/facilitates whole team in story delivery
- Facilitates meetings
- Optimizes process
- Coordinates project work streams
- Helps manage project/iteration scope
- Manages external dependencies
- Escalates risks/issues
- Reports on progress

Developers


0110001011001101100110110111

Developer

- Turns user stories/requirements into working software
- Conducts the technical analysis of requirements
- Assists in the evolution of user experience designs
- Coordinates the implementation of supporting architecture/infrastructure
- Assists with testing
- Estimates user stories
- Works with QA to automate testing scenarios
- Presents work to QA and BA for final testing
- Escalates risks and issues that impact scope and timeline
- Writes code in line with project standards
- Stands behind architectural decisions, patterns and best practices that the team has agreed to

Tech lead

Tech lead

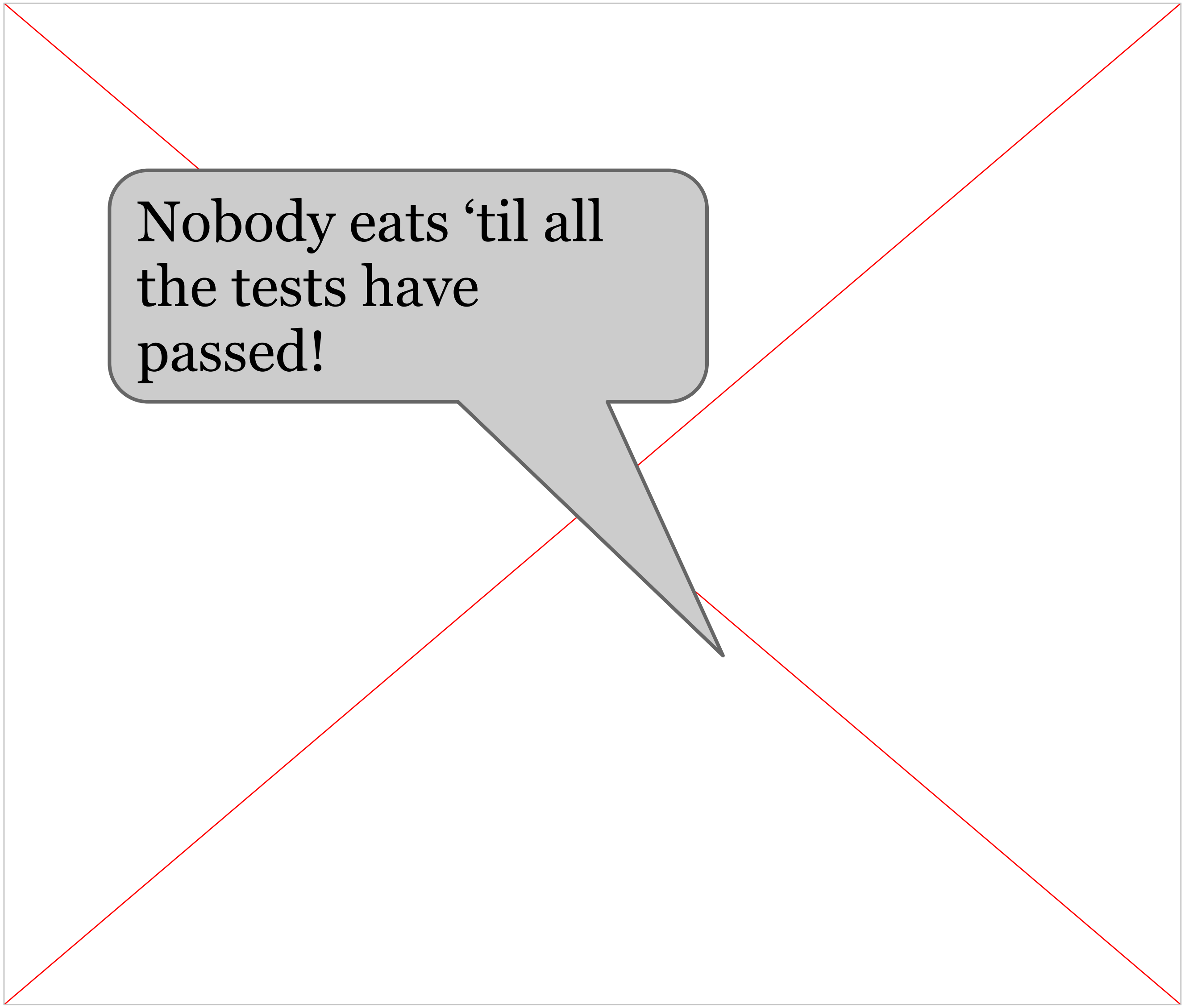
- Assists in planning activities and representing the engineering team at planning meetings when the rest of the team is not present
- Owns the maintenance of the quality and architectural coherence of the code developed by the project
- Escalates significant architectural/technical issues for prompt resolution
- Coordinates regular refactoring/code hygiene to prevent accumulation of technical debt
- Communicates, educates, and enforces best practices and standards
- On-boards new team members
- Bi-Weekly sync up with ops

UX/UI Designer

UX/UI Designer

- Gathers data to assess needs/requirements of users
- Proposes designs to define how users will interact with the functionality of the product (including designs of UX in general, and the product's interface in particular)
- Provides design expertise and guidance to engineers and QA during code writing and testing
- Ensures that the product is not only useful, but usable as well
- Assists in narrative and user story development, particularly in the delivery of development-ready design assets and/or prototypes
- Leads usability testing and logs associated findings

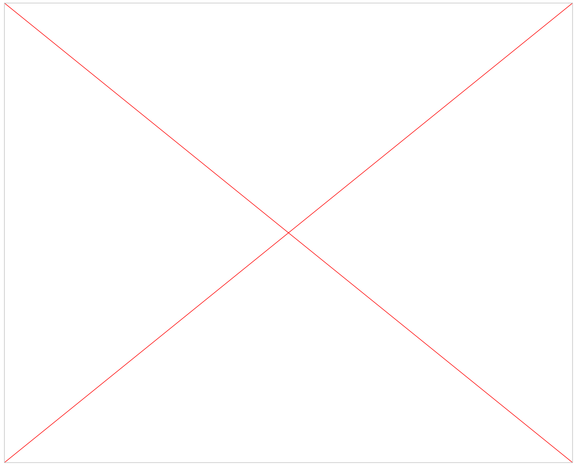
Quality Assurance Analyst



QA Analyst

- Produces test cases/scenarios
- Manually tests when not automated/automatable
- Maintains regression test suite
- Integration testing
- Exploratory testing
- Assists in acceptance criteria definition
- Trains and otherwise works with engineers in best-practices for assuring code and product quality

Business Analyst



Business analyst

- Maintains story pipeline
- Advises product owner/managers on story priorities
- Assists in definition of acceptance criteria
- Communicates requirements to engineers and user experience designers
- Helps control scope of iterations and the project as a whole
- Showcases the product
- Testing stories for completion

Questions?

Next steps:

- Collectively determine who will own the various responsibilities
- Wikimania agile panel “Hacking our teams: Flexible ‘agile’ development at the WMF”
- Mobile web team documentation: http://bit.ly/11uUyXt
- Mingle instance: http://bit.ly/14iH0lP

Contact info

- Tomasz Finc, tfinc@wikimedia.org
- Arthur Richards, arichards@wikimedia.org
- #wikimedia-mobile