

Open Project Management

from an “open” perspective

UNIT 4

Instructor: Dr. Bradly Alicea

<http://bradly-alicea.weebly.com>



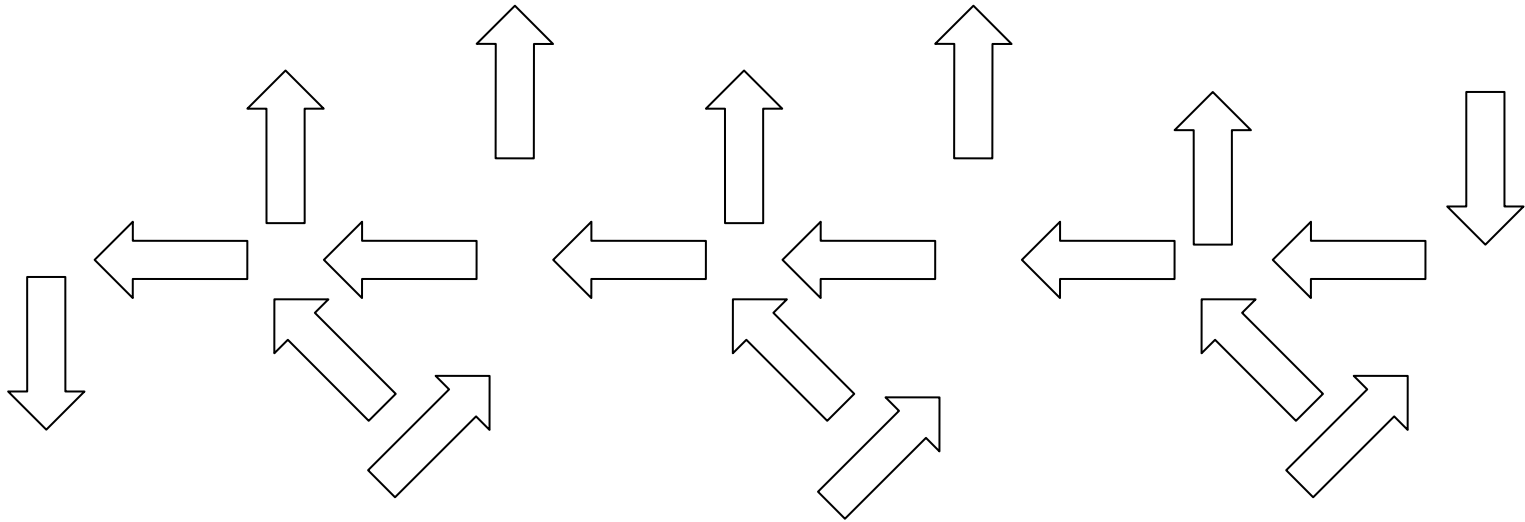
Lecture 15

All content



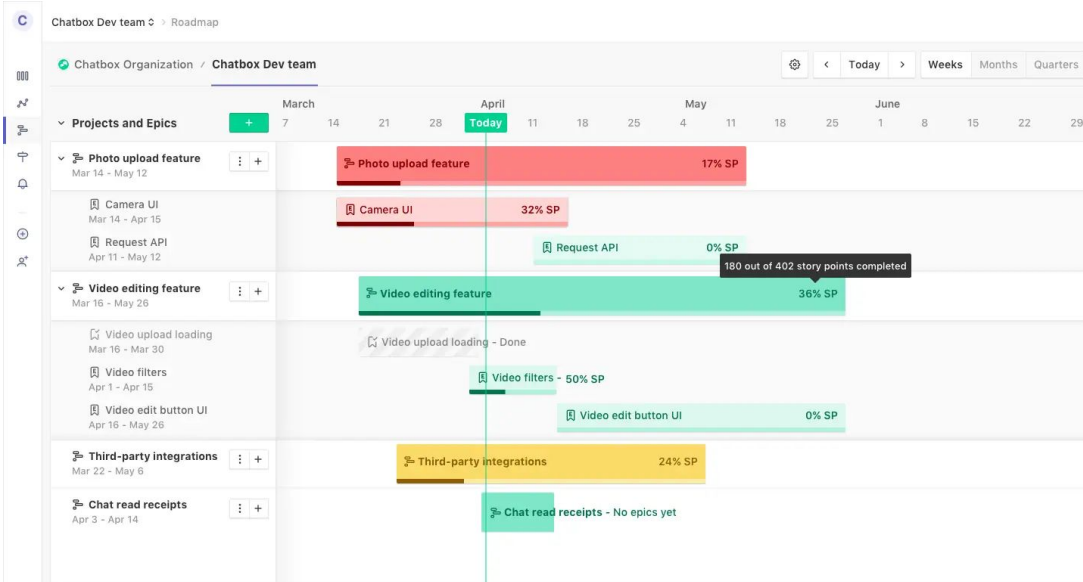
Open Project Management

Welcome!



Project Scope and Types of Contribution

Zenhub Roadmaps and Epics



Code Issues 86 Pull requests ZenHub Actions Wiki

Repos (2/2) Labels Milestones Assignees Epics Releases Estimates

1 Issue - 0 Story Points
Icebox

8 Issues - 18 Story Points
Epic

6 Issues - 19 Story Points
Priority Epics

ZenHub-Demo #418
markdown enhancements
Enhancement

ZenHub-Demo #171
API Interface
Sprint 45
Client Project 1
Webinar Release Project
New Feature A
Demo Release
Filter by Epic Issues

Blocked: Design Design
User Story

ZenHub-Demo #340
Dashboard updates
Sprint 45
Filter by Epic Issues

Epic

Getting-Started-Demo #7
UI Updates and Bugs
Sprint 45
Test Epic ZenHub 101
Release 1
Sample Release IBM Demo
Filter by Epic Issues

ZenHub-Demo #134
credit card payments

High priority
ZenHub-Demo #22
Dashboard updates
Sprint 50
Sample Epic
Sample Title for Shane
Q2 - Project Planning: Testing...
Q2 - New Payment System
Payments Release
Webinar Release Project
Demo Release
New Feature A
Filter by Epic Issues

45mins Epic Feature
High Priority

High priority
ZenHub-Demo #135
Create a new payments
dashboard
Sprint 45
Q3 - Project Epic
Epic X
Q2 - New Payment System
Payments Release
Webinar Release Project
New Feature A
Filter by Epic Issues


<https://www.zenhub.com/>

Roadmaps involve defining deliverables, identify risks, milestones, and key resources.

Roadmaps involve defining deliverables, identify risks, milestones, and key resources.

Deliverable: what you will produce at the end of your project or unit.

Deliverable: what you will produce at the end of your project or unit.



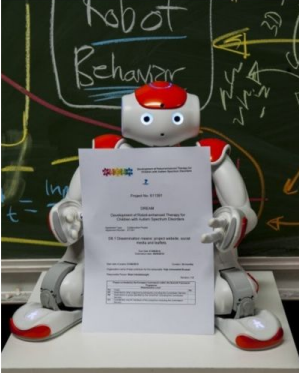
Development of Robot-Enhanced therapy for children with Autism spectrum disorders

Goals & Methodology | Publications | Platform

Consortium | Deliverables | About

Download ZIP File | Download TAR Ball | View On GitHub

Deliverables



Number	Name	Lead	Date
D1.1	Intervention Definition	UBB	23/06/2014
D1.2	Robot Behaviour Specification	UBB	07/07/2014
D1.3	Child Behaviour Specification	UBB	07/07/2014
D1.4	Manual of best practice in robot enhanced therapy for autism spectrum disorder	UBB	10/05/2019
D2.1.1	Tasks for social robots on developing social skills (Wizard of Oz system)	UBB	06/04/2015
D2.1.2	Tasks for social robots on developing social skills (Wizard of Oz system)	UBB	08/09/2015
D2.2.1	Tools for the assessment of child-robot interaction and diagnostics	UBB	01/04/2017

The DREAM project finished in March 2019. This site is a static version of the former project website: dream2020.eu, which is no longer actively maintained.

Hosted on GitHub Pages — Theme by [orderedit](#)

<https://dream2020.github.io/DREAM/deliverables.html>

Roadmaps involve defining deliverables, identify risks, milestones, and key resources.

Deliverable: what you will produce at the end of your project or unit.

Milestones: what things need to be completed before other things can be completed?

Soup: open can → pour into pan → turn on stove → stir → serve

Milestones: what things need to be completed before other things can be completed?

Soup: open can → pour into pan ~~X~~ → turn on stove → stir → serve

Milestones: what things need to be completed before other things can be completed?

Soup: open can → pour into pan ✗ → turn on stove → stir → serve



Roadmaps involve defining deliverables, identify risks, milestones, and key resources.

Deliverable: what you will produce at the end of your project or unit.

Milestones: what things need to be completed before other things can be completed?

Soup: open can → pour into pan → turn on stove → stir → serve

Identify Resources and Risks: what needs to be in place before you can complete your project or unit? What the the risks of pushing back a deadline, or of making certain design choices?

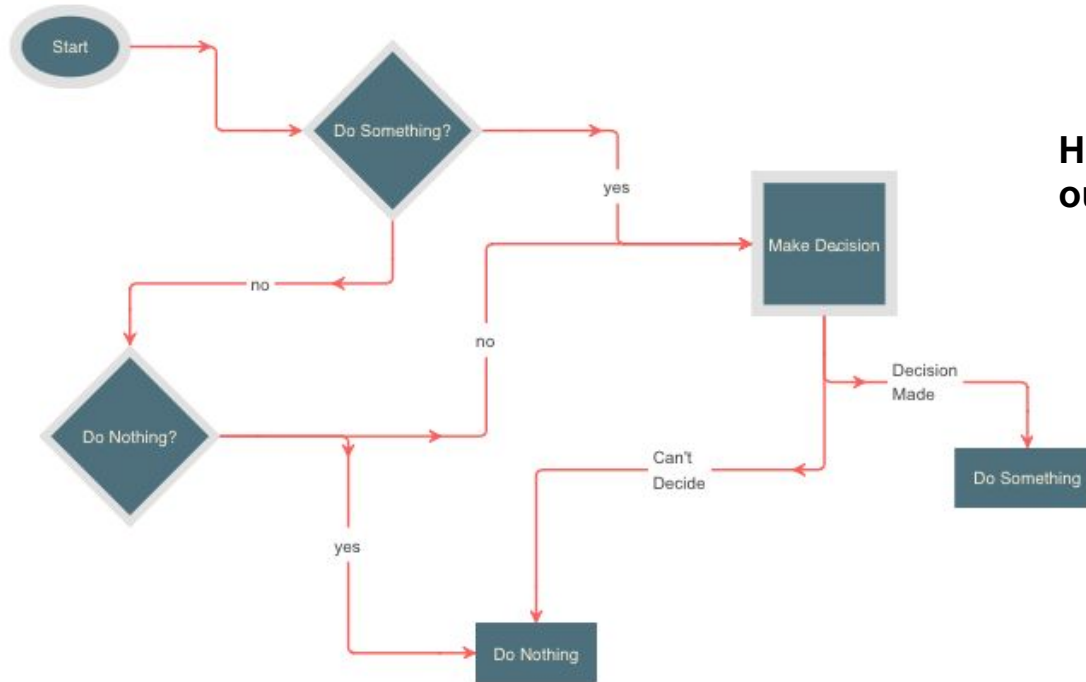
Issue Creation and Tradeoffs



Claude Shannon: Information Theory, Robotic Agents, Juggler?!?

<http://lkozma.net/blog/shannons-juggling-theorem/>

Roadmap: issues at the systems-level



How do we make a set of issues out of this roadmap?

COURTESY

<https://www.goodfirms.co/workflow-management-software/blog/best-free-open-source-workflow-management-software>

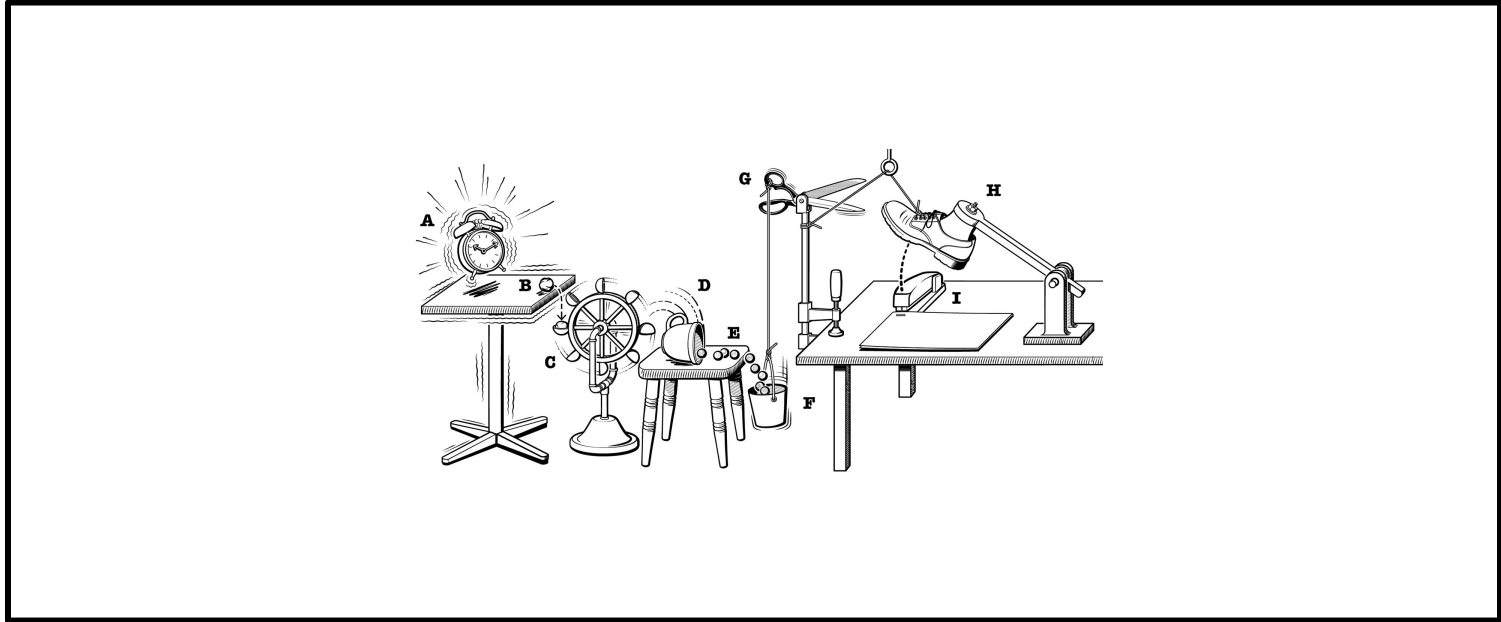
Issue: what is the optimal size?

What do you want to achieve?



Issue: what is the optimal size?

What do you want to achieve?



What dependencies are involved?

Issue: what is the optimal size?

What do you want to achieve?



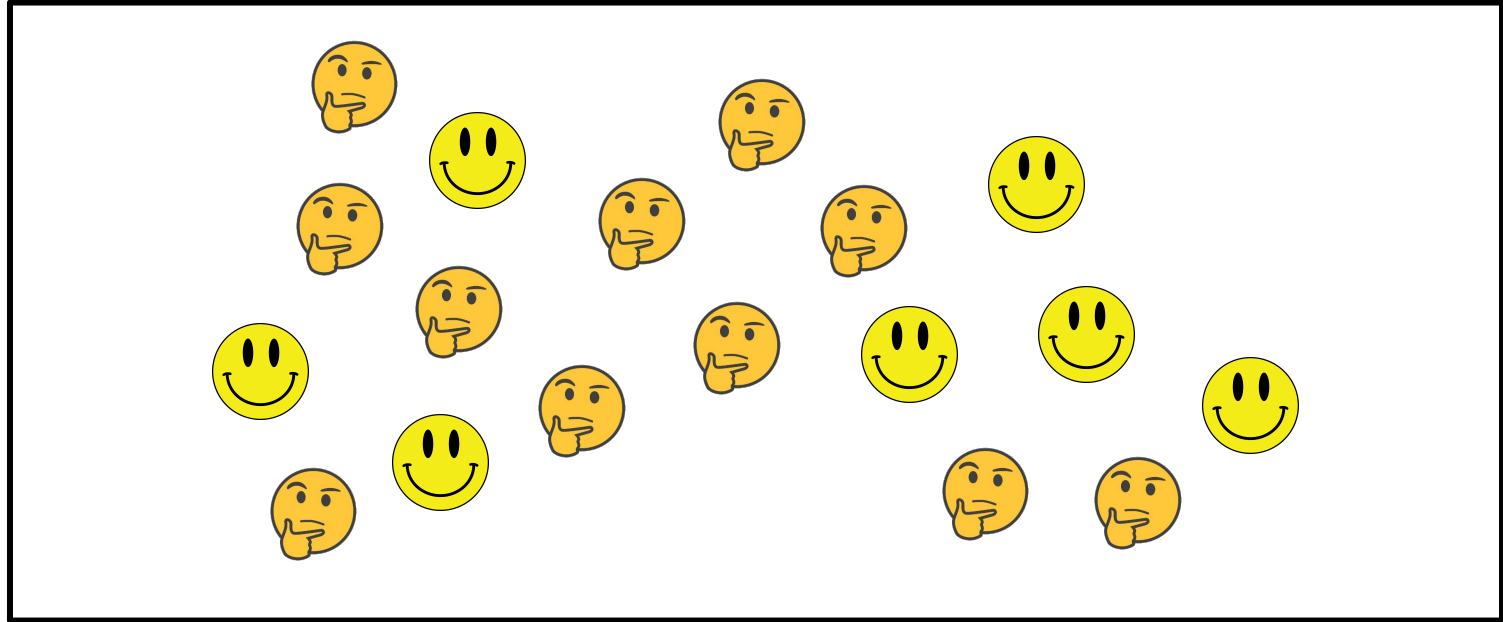
What dependencies are involved?

What is your time horizon?

Issue: what is the optimal size?

What do you want to achieve?

Who (expertise) is involved?



What dependencies are involved?

What is your time horizon?

Issue: what is the optimal size?

Issues are a consolidation of communication channels and interacting project goals and needs.

- a “project” is actually a program: a program of related projects moving in the same direction.
- but these subprojects interact and conflict: goal is to optimize the flow of activity and enable new ideas.

The fewer contributors your organizations and projects has, the less “process” you will need.

- fewer issues, but also less focus on issue creation and refinement.
- number of communication channels in a team goes up as the number of people on your team goes up.

Issue: what is the optimal size?

What do you want to achieve?

Project vision, break down vision into smaller pieces.

- what to do first, then next, then even farther in the future. Order of plausibility, priority, maturity.

	Plausibility	Priority	Maturity
Issue A			
Issue B			
Issue C			

Issue: what is the optimal size?

What do you want to achieve?

Project vision, break down vision into smaller pieces.

- what to do first, then next, then even farther in the future. Order of plausibility, priority, maturity.

	Plausibility	Priority	Maturity
Issue A	HIGH	HIGH	EARLY
Issue B	MED	HIGH	MIDDLE
Issue C	LOW	LOW	LATE

Issue Attributes

Size, **Scope**, and **Scale**

Size: what is the amount of work that needs to be done? Rule of thumb: how many words does it take to define issue?

Issue Attributes (con't)

Size, **Scope**, and **Scale**

Size: what is the amount of work that needs to be done? Rule of thumb: how many words does it take to define issue?

Scope: what is the depth of complexity? Are there a lot of dependencies, or does it require new things to be put into place?

Issue Attributes (con't)

Size, **Scope**, and **Scale**

Size: what is the amount of work that needs to be done? Rule of thumb: how many words does it take to define issue?

Scope: what is the depth of complexity? Are there a lot of dependencies, or does it require new things to be put into place?

Scale: how long does it take to complete your issue?

Issue Attributes (con't)

From Bug Tracking (Chapter 10) in “Program Management for Open source Projects”
(Ben Cotton)

Problem: define problem (e.g. build a library of documents).

Type/Version: label/description (e.g. getting started) and iteration (v1.0.2).

Compositionality: part of a larger component (e.g. linked to curation).

Status/Priority/Impact: indication of importance and order of importance (e.g. urgent).

Issue Attributes (con't)

Closure Criterion: what needs to be completed before an issue can be closed?

- Who needs to review and approve? Who needs to be aware of changes?
- Public review of issues (monthly) can solve this.

Issue Attributes (con't)

Closure Criterion: what needs to be completed before an issue can be closed?

- Who needs to review and approve? Who needs to be aware of changes?
- Public review of issues (monthly) can solve this.

Who's responsible?

- Who makes sure that the status of all issues is kept up to date?
- Maintainer, code owner, person who creates (or is tagged in) the issue.

Issue Attributes (con't)

Closure Criterion: what needs to be completed before an issue can be closed?

- Who needs to review and approve? Who needs to be aware of changes?
- Public review of issues (monthly) can solve this.

Who's responsible?

- Who makes sure that the status of all issues is kept up to date?
- Maintainer, code owner, person who creates (or is tagged in) the issue.

Place in timeline: how do issues get prioritized?

- first in, first out principle: oldest issues get resolved first.
- contingencies first, dependencies next: issues in order of necessity.