

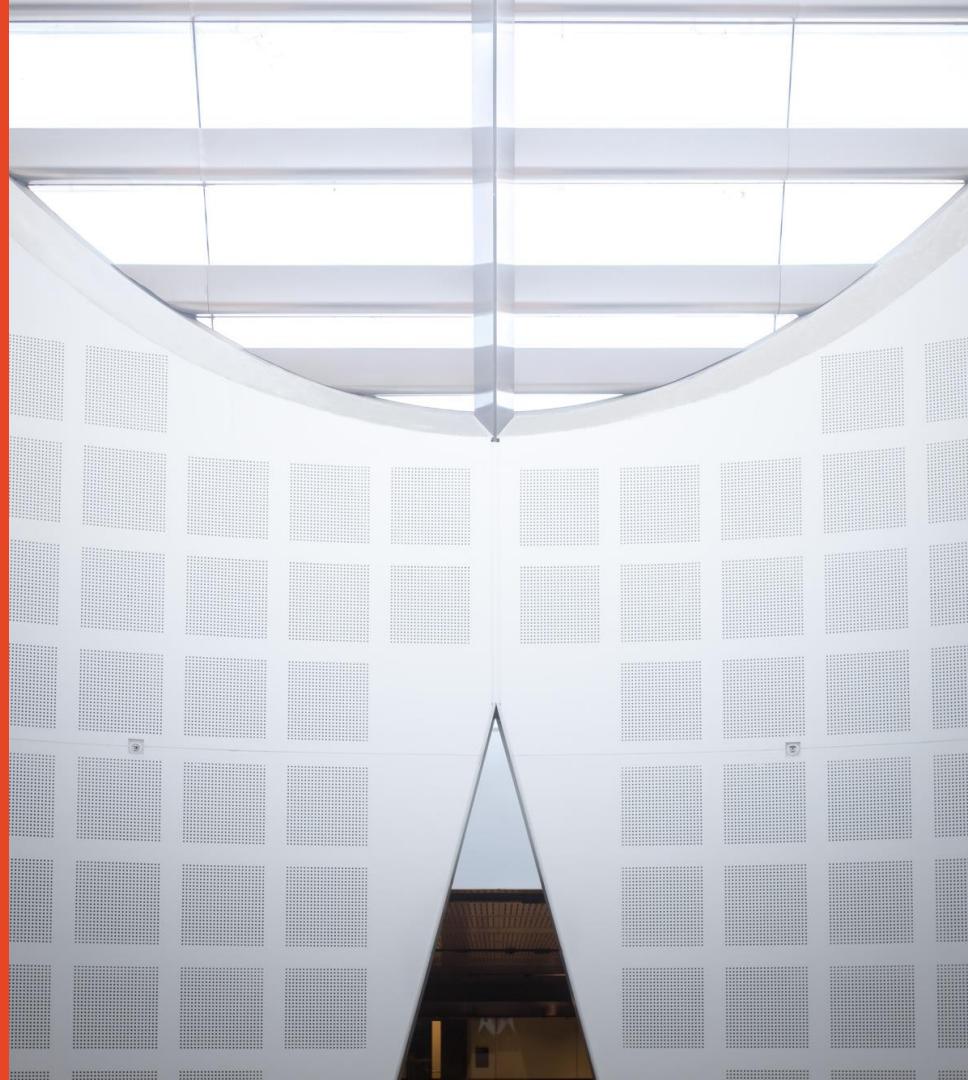
Agile Software Development Practices

SOF2412 / COMP9412

**Team Dynamics; Tools and
Technologies for Teamwork**

Dr. Basem Suleiman

School Computer Science

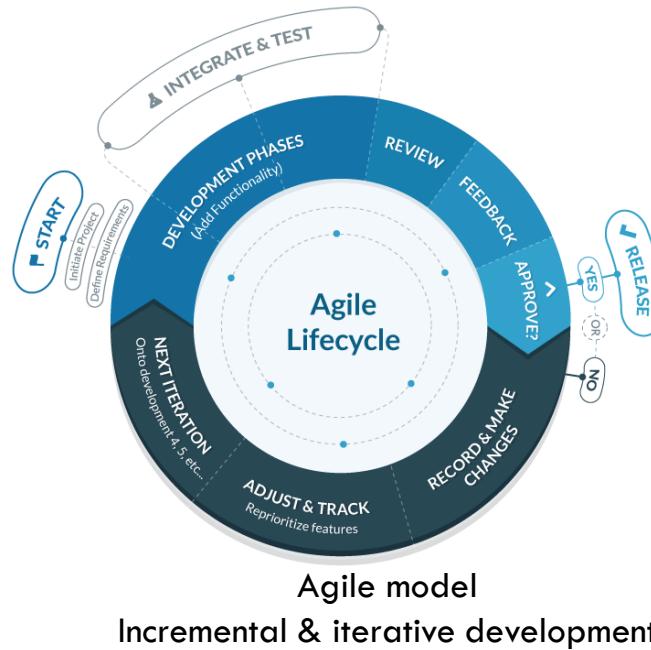
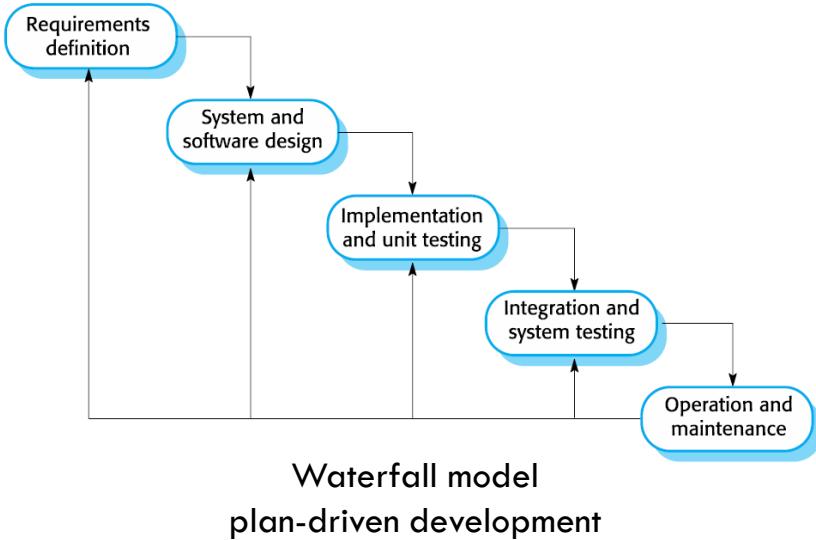


Agenda

- Teams in Software Development Models
- Teams in Agile Development
- Team Dynamics
- Effective Teams and Teamwork
- Issue Tracking
- Tools for Issue Tracking and Teamwork

Software Development Models – Teams

Do software development models influence team structure and interactions in software development?



<https://blog.capterra.com/agile-vs-waterfall/>

Waterfall Model – Teams

Development activities	Teams
Divide the work into stages	A separate team of specialists for each stage
At each stage, the work is passed from one team to another	Some coordination is required for the handoff from team to team – using “documents”
At the end of all of the stages, you have a software product ready to ship	As each team finishes, they are assigned to a new product

Teams under different SDLC models

- In a traditional structure how do teams work?
 - As work is planned and allocated, it can be divided into pieces that should be more-or-less independent
 - Specialist teams
 - Project management and resource reallocation
 - Clear authority lines, so disagreements can be resolved
 - Problems?
 - Single points of failure
 - Inflexibility
 - Lack of feedback
- And in Agile teams?

Agile Manifesto – Revisit

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

Agile Manifesto –

- **Individuals and interactions over processes and tools**
- Why Agile values individuals and interactions over processes and tools? Discuss

Agile Manifesto – Why Individuals and Interactions?

- Why Agile values individuals and interactions over processes and tools?
 - People tend to follow processes blindly, and make mistakes
 - “A great tool can sometimes help people to do the wrong thing faster”
 - Tools or best practices are not enough - people who need to use it should buy into it to realize its benefits
 - People needs to see the value of following certain practices
- It is important to recognize that you are working with a group of people who have different motivations, ideas and preferences

Agile Principles 1 – People

- Build projects around **motivated individuals**. Give them the environment and support they need, and **trust** them to get the job done
- The most efficient and effective method of **conveying information** to and within a development team is **face-to-face conversation**
- At regular intervals, the **team reflects** on how to become more **effective**, then **tunes and adjusts its behavior** accordingly

Agile Principles 1 – People

- *Group Discussion:*
 - *Briefly discuss how the following agile principles perceive teams and teamwork in agile development?*
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Agile Principles 2 – People

- Business **people** and **developers** must **work together daily** throughout the project
- The best architectures, requirements, and designs emerge from **self-organizing teams**
- Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software

Agile Principles 2 – People

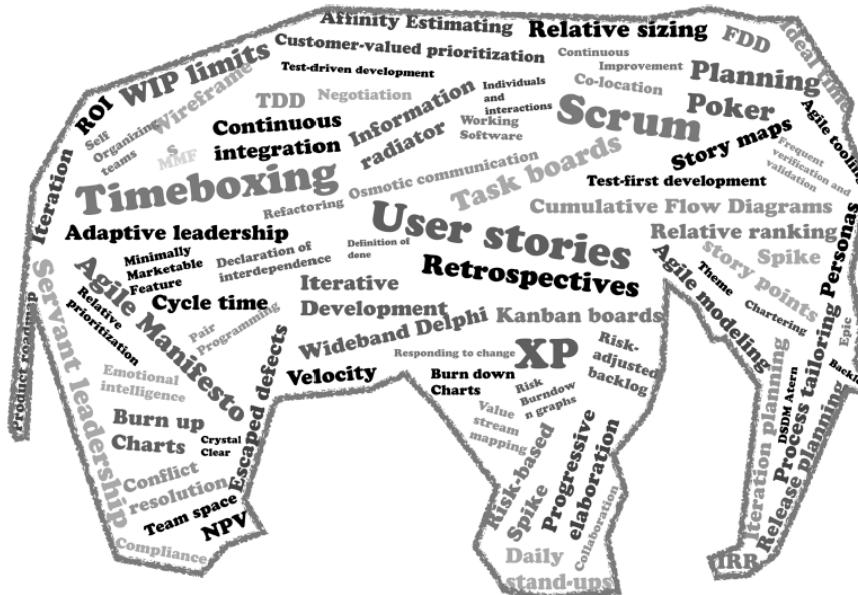
- *Group discussion*
 - *briefly discuss how the following agile principles perceive teams and teamwork in agile development?*
- Business people and developers must work together daily throughout the project
- The best architectures, requirements, and designs emerge from self-organizing teams
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software

Teams – Individuals and Collaboration

- Common problem experienced in software development teams “*throw it over the wall*”
 - Team members are busy thinking about their own project work and problems
 - Different views/perspectives
 - Teams are divided, and collaboration is killed

Agile Practices – The Agile Elephant

- The agile elephant is made up of many practices



*Agile practices and Scrum method (most of these presented in this figure) will be discussed in more details in future lectures. The focus of the discussion here is on teams and the adoption of Agile practice

Agile Teams – Individual Practices

- When adopting agile practices, team members may adopt practices individually:
 - Each person uses only the practices that impact their work; developers focus on automated tests and build, team leads on task boards, project velocity and burn-down charts, business users on user stories
 - Adopting practices individually will improve things, but this may lead to a self-contradictory effect

Teams – Individual Practices

- When adopting agile practices, team members may follow the same thinking:
 - Each person uses only the practices that impact their work; developers focus on automated tests and build, team leads on task boards, project velocity and burn-down charts, business users on user stories
 - Adopting practices individually will improve things, but this may lead to a self-contradictory effect
 - Each person sees the part of agile that affects their specific work – (attitudes: “see! I was right all along”)
 - Agile is made up of day-to-day practices, but it's much bigger than those practices

Understanding the Agile Elephant

- If you only sees the practices that directly affect your project work, then you will see the one small piece of agile

Not the whole elephant!



Team Dynamics



THE UNIVERSITY OF
SYDNEY



Team Dynamics

- “Team dynamics are the unconscious, psychological forces that influence the direction of a team’s behaviour and performance”
- Factors that lead to team dynamics:
 - Personalities and work styles
 - Knowledge and skills
 - Organization culture and structure
 - Cultural differences, background

Team Dynamics – Good or Bad?

In Agile development, is team dynamics a good or bad thing?
Discuss

Team Dynamics – Pros and Cons

- Can be good
 - E.g., Improve overall team performance (productive conflict, different perspectives)
- Can be bad
 - Can lead to unproductive conflict can demotivate and prevent team from achieving its goals

Team Dynamics – Tuckman Team Development Model



Image: <https://www.atlassian.com/agile/teams>

Tuckman's stages of group development - https://en.wikipedia.org/wiki/Tuckman%27s_stages_of_group_development

Team Dynamics – Identification and Resolution

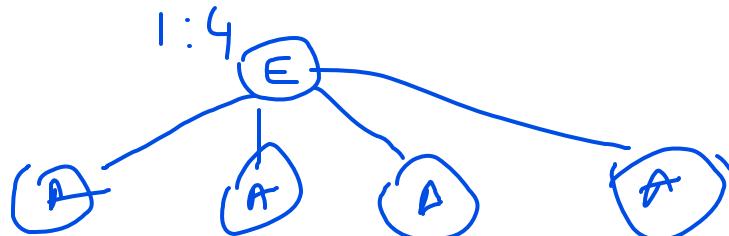
- Result from the interaction of many factors
 - E.g., Personalities, work style, roles, culture, organizational structure
- Investigate the root causes of conflict or poor team performance
 - Structured interviews or informal chats in a private and confidential
- Identify potential improvements
 - E.g., change in office layout, team development workshops (practices,
3 personality dynamics, cultural change programs)
1 2

Source: <https://mysoftwarequality.wordpress.com/2014/09/04/cross-dysfunctional-teams>

Agile Teams – Skills

- It's claimed that agile teams work with the best developers. However, this is not necessarily the case;
 - Every project needs at least one experienced and competent lead person (Critical Success Factor)
 - Each experienced and competent person on the team permits the presence of 4-5 “average” or learning people
 - With that skill mix, agile techniques have been shown to work many times

Skill mix
preference



E: experienced ♂
A: average ♀

Teams Culture

- Consider the following scenarios:
 - Developer: “It works in our environments, it’s operations responsibility to make it work in production”
 - Tester: ”Listen, it worked in User Acceptance Testing, it must be a configuration issue, or a missing firewall hole and nothing I could have spotted during testing...”
 - Customer: “Hello! Nothing works here...”
- Are these statements signs of team dynamics?
- Should this kind of culture exist in agile teams?

*Yes, differing views
No!*

Source: <https://mysoftwarequality.wordpress.com/2014/09/04/cross-dysfunctional-teams>

Effective Teamwork

- Teamwork comprises of the right tools, the right people ad the right practices
- Effective teamwork is everyone's shared responsibility
- Large software organizations, teams involve many roles across different departments (engineering, design, sales/marketing, legal)
- Use team building activities to build effective teams

Team Building

- “Various types of activities used to enhance social relations and define roles within teams, often involving collaborative tasks”
- There is an evidence how team building affect positively team effectiveness

Team Building Activities

- **Goal setting:** emphasizes the importance of clear objectives and individual and team goals
 - E.g., objectives and key results (Atlassian)
- **Interpersonal relations:** focus on teamwork skills such as giving and receiving support, communication and sharing information
- **Standup meetings**
- **Roles and responsibilities** (Atlassian)

Team Building – Roles and Responsibilities

- Define the roles and responsibilities that will make your team successful
- Clarify expectations as a team
- Helps to move a team from “storming” to “norming”, or help “performing” teams to get back on track

Roles and Responsibilities – How?

- Create a table of roles and responsibilities
 - Responsibilities from own perspective
 - Responsibilities from team member's perspective

Other's perspective
P

T

Name	Role	Responsibility (own)	Responsibility (other's)
CD	PO	Backlog	Backlog + communicating with Client etc ..
.	.	.	.
.	.	.	.

Roles and Responsibilities – How?

- Identify roles
 - E.g., team lead, developers, designers
 - Coarse-grained
 - Add to the role's column
- Clarify own responsibility
 - Think of top 3-5 tasks in priority order
 - Write on sticky notes
- Think of teammate's responsibilities
 - Write 1-2 responsibilities for each role from your perspective
 - Write responsibilities you may think that don't have a clear owner

Roles and Responsibilities – How?

- Refine and consolidate (optional)
 - Talk teammates with similar roles and refine responsibilities
- Discuss all roles
 - Role owner(s) describe their role and place their sticky notes in own responsibility column
 - Other teammate's role description in the other column
 - Owner to accept/decline the responsibilities by other teammates (suggest role to own it). Define primary owner for overlapping roles
 - Add “unassigned responsibilities” to
- Summarize roles and responsibilities
 - All to agree
 - Owner to document it and how to fill skill gaps

High-Performing Agile Teams

- Cross-functional; engineers, designers, architects, sales
- Mutual respect and mutual responsibility
 - Not blaming culture, and/or “throw it over the wall”
- Sound engineering practices (tools and automation)
- Value and belief of agile practices and principles
- Apply agile practices effectively as individuals and as a team
- Receive continuous training (technical and non-technical) and team monitoring/coaching

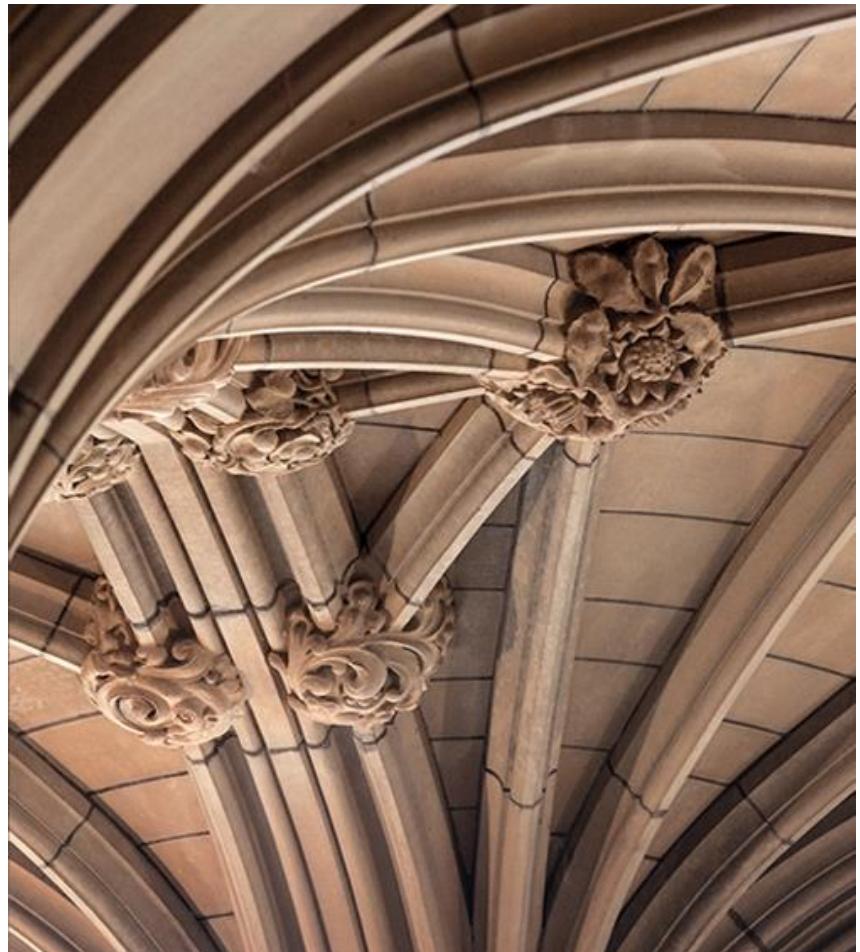
How much can you find out?

- Search for:
 - Team effectiveness
 - Self-managed teams
 - Group conflict
 - Team efficacy...
- Learn about the theory of teams!

Tools and Technologies for Teamwork



THE UNIVERSITY OF
SYDNEY



Issue Tracking Systems

- A software that manages and maintains lists of issues
- Used to create, update and resolve reported issues internally or externally
- **Bug (defect) Tracking System:** keeps track of reported software bugs in software development projects
 - Centralized overview of development requests and their states
 - May assign a priority, status, severity and/or complexity
 - Prioritized list of pending items (Backlog)
 - Typically integrated with other tools or software management systems

https://en.wikipedia.org/wiki/Issue_tracking_system
https://en.wikipedia.org/wiki/Bug_tracking_system

Trello, github issues etc..

Bug/Issue Tracking Part of other Systems

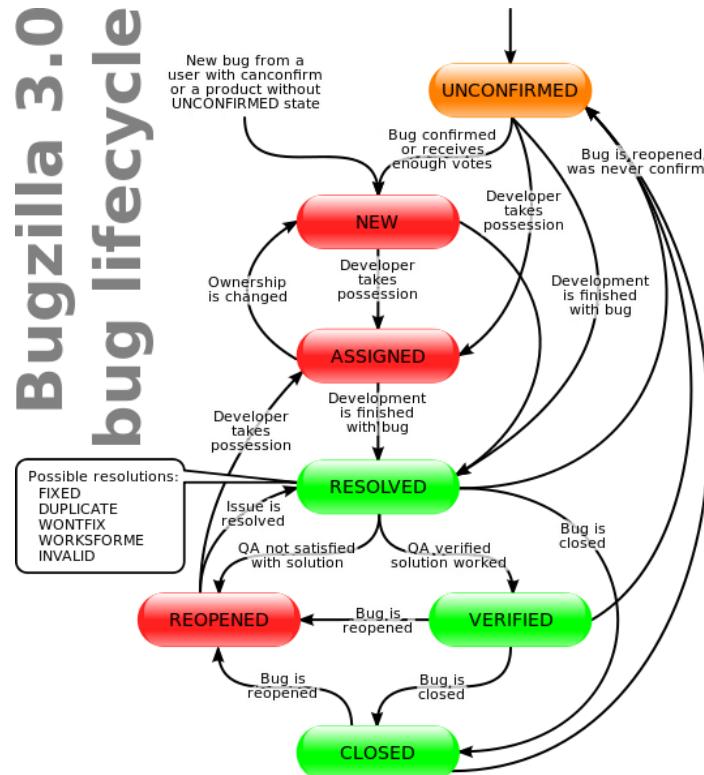
- Part of integrated project/software development management systems
- It helps integrating issue/bug tracking with other activities
- Distributed bug tracking tools are designed to be used with distributed revision control software

Bugzilla – Bug Tracking Tool

- Open-source web-based bug tracker and testing tool by Mozilla project
- Bug (or feature) requests can be submitted by anyone and will be assigned to a particular developer
- Various status updates for each bug
 - E.g., Bugzilla itself allows the public to file bugs – it assigns all bugs to a gatekeeper whose job is to assign responsibility and priority level

<https://en.wikipedia.org/wiki/Bugzilla>

Bugzilla – Bug Lifecycle



By Nyco [GFDL (<http://www.gnu.org/copyleft/fdl.html>) or CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>)], from Wikimedia Commons

Issue Tracking – GitHub

SOFT2412-Agile-Software-Development / Test Private

Watch 0 Star 0 Fork 0

Code Issues 2 Pull requests 0 Projects 1 Wiki Insights Settings

 Login bug

Write Preview

When a user enters a username or password with non-permitted characters such as a semicolon

Assignees
bsul6138
asan0483

Labels
bug

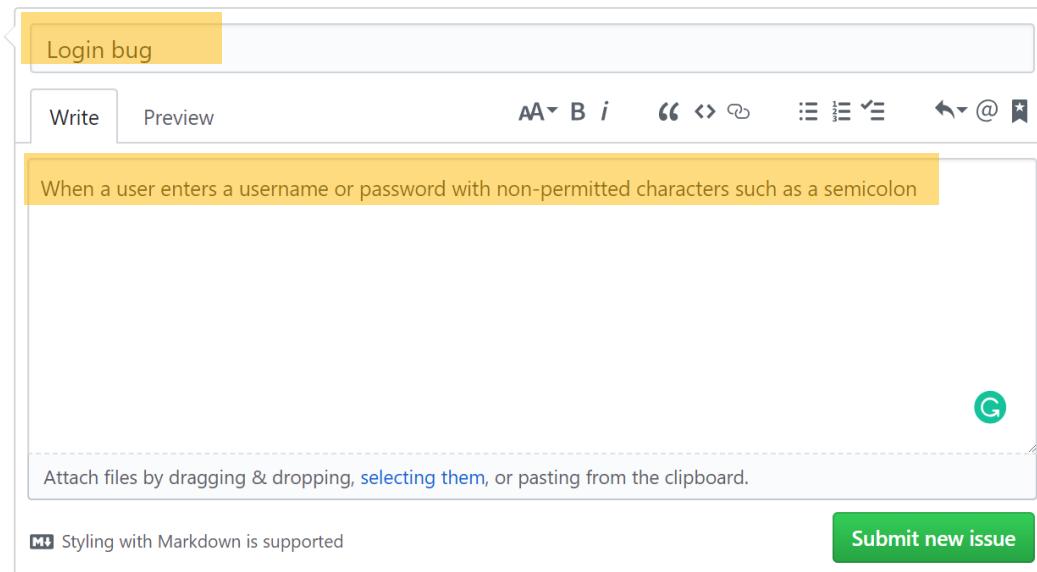
Projects
Test repo

Milestone
No milestone

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Styling with Markdown is supported

Submit new issue



GitHub Issue Tracking – Edit Labels

SOFT2412-Agile-Software-Development / Test Private

Watch 0 Star 0 Fork 0

Code Issues 2 Pull requests 0 Projects 1 Wiki Insights Settings

Search all labels Labels Milestones New label

8 labels			Sort ▾
bug	Something isn't working	Edit	Delete
duplicate	This issue or pull request already exists	Edit	Delete
enhancement	New feature or request	Edit	Delete
good first issue	Good for newcomers	Edit	Delete
help wanted	Extra attention is needed	Edit	Delete
invalid	This doesn't seem right	Edit	Delete
question	Further information is requested	Edit	Delete
wontfix	This will not be worked on	Edit	Delete

GitHub Issue Tracking – Project Management

The screenshot shows the GitHub Projects interface for the repository "SOFT2412 - Agile Software Development Practices". The interface is divided into three main columns: "To do", "In progress", and "Done".

- To do:** Contains 5 items:
 - "Enter a note" (new card)
 - "Search filter user story" (added by bsul6138)
 - "Product Search User Story" (added by bsul6138)
 - "Automation" (description: "Automatically move your cards to the right place based on the status and activity of your issues and pull requests.", added by ffar6831)
 - "Welcome to GitHub Projects" (description: "We're so excited that you've decided to create a new project! Now that you're here, let's make sure you know how to get the most out of GitHub Projects.", added by ffar6831)
- In progress:** Contains 1 item:
 - "Test 2 - added" (added by hoso5448)
- Done:** Contains 2 items:
 - "test" (status: "Test#2 opened by ffar6831")
 - "Complete User Sign-up" (status: "Test#3 opened by bsul6138")

At the top, there are navigation links for "Pull requests", "Issues", and "Explore". Below the columns, there is a search bar labeled "Filter cards", a button to "Add cards (1 new)", and options for "Fullscreen" and "Menu". A red box highlights the "To do" column, and another red box highlights the "Done" column.

Version Control Systems – GitHub Revisit

- GitHub allows groups of people to collaborate across many projects at the same time in organizations account
- Organization's members can be:
 - Owner: complete administrative access to the organization
 - Member: default role for everyone else
- Owners can manage members' access to the organization's repos. and projects with fine-grained permission controls
- Can add collaborators from outside of the organization (consultant) to have access to one or more organization repos. without being a member of the organization

GitHub – Organization Access Control (Revisit)

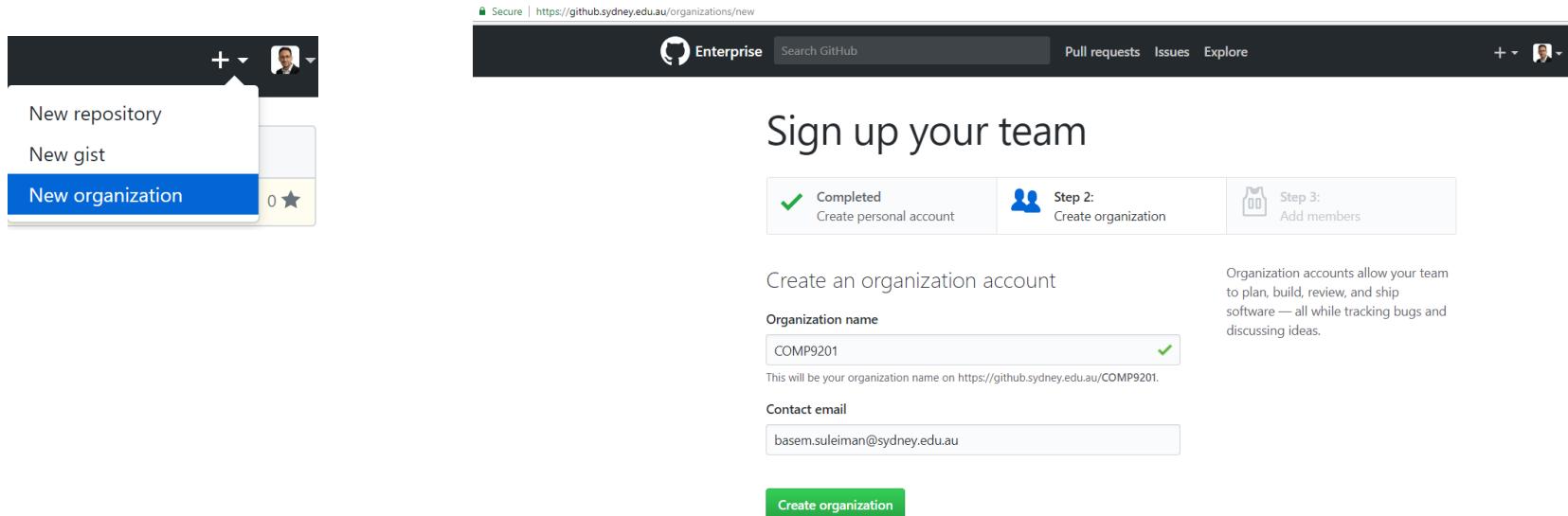
Organization action	Owners	Members
Invite people to join the organization	X	
Edit and cancel invitations to join the organization	X	
Remove members from the organization	X	
Reinstate former members to the organization	X	
Add and remove people from all teams	X	
Promote organization members to <i>team maintainer</i>	X	
Add collaborators to all repositories	X	
Access the organization audit log	X	
Delete all teams	X	
Delete the organization account, including all repositories	X	

Organization action	Owners	Members
Create teams	X	X
See all organization members and teams	X	X
@mention any visible team	X	X
Can be made a <i>team maintainer</i>	X	X
Transfer repositories	X	
View a project board and add or reorganize its cards and columns	X	X
Create or delete a project board and edit its description	X	X
Automate actions for project boards	X	X
View and post private team discussions to all teams (see " About team discussions " for details)	X	
Edit and delete team discussions in all teams (for more information, see " Managing disruptive comments	X	

- Examples of access permissions for organization's owners and members

<https://help.github.com/enterprise/2.13/user/articles/permission-levels-for-an-organization/>

GitHub – Creating Organization (Revisit)



The screenshot shows the GitHub interface for creating a new organization. On the left, a sidebar menu includes options for 'New repository', 'New gist', and 'New organization'. The 'New organization' option is highlighted with a blue background and white text. The main content area has a title 'Sign up your team' and a progress bar indicating 'Step 1: Completed Create personal account', 'Step 2: Step 2: Create organization', and 'Step 3: Step 3: Add members'. Below the progress bar, there are fields for creating an organization account: 'Organization name' (set to 'COMP9201'), 'Contact email' (set to 'basem.suleiman@sydney.edu.au'), and a 'Create organization' button.

Secure | https://github.sydney.edu.au/organizations/new

Enterprise Search GitHub Pull requests Issues Explore +

New repository New gist New organization 0 ★

Sign up your team

Completed Create personal account Step 2: Create organization Step 3: Add members

Create an organization account

Organization name

COMP9201 ✓

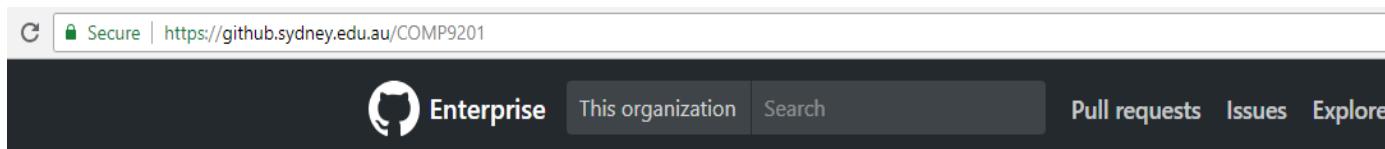
This will be your organization name on https://github.sydney.edu.au/COMP9201.

Contact email

basem.suleiman@sydney.edu.au

Create organization

Organizational accounts have a namespace where all their projects exist



The screenshot shows the main page of the GitHub organization 'COMP9201'. The header includes the GitHub logo, the text 'Enterprise', and navigation links for 'This organization', 'Search', 'Pull requests', 'Issues', and 'Explore'. The URL in the address bar is 'https://github.sydney.edu.au/COMP9201'.

Secure | https://github.sydney.edu.au/COMP9201

Enterprise This organization Search Pull requests Issues Explore

GitHub – Add Members to Organization (Revisit)

The screenshot shows the GitHub Enterprise interface with the following details:

- Header: Enterprise, This organization, Search, Pull requests, Issues, Explore.
- Section title: Add organization members.
- Step navigation: Completed (Create personal account), Step 2: Create organization, Step 3: Add members (highlighted).
- Search bar: Search by username, full name or email address (with 'hos05448' typed in).
- Result list: hos05448 (Hamzah Bin Osop) and Faraz Farid.
- Buttons: Finish.
- Right sidebar: Organization members (See all repositories, Create repositories, Organize into teams, Review code, Communicate via @mentions). A note for organization owners about access rights.

- Note: when you create a new repo you can create them under your personal account or under any of the organizations that you're owner in

The screenshot shows the GitHub interface for creating a new repository:

- Section title: Create a new repository.
- Description: A repository contains all the files for your project, including the revision history.
- Form fields:
 - Owner: COMP9201 (selected).
 - Repository name: WebStoreApp (checked).
- Dropdown: Choose another owner (bsul6138, COMP9201 selected).
- Text area: Need inspiration? How about [list of suggestions: INFO3220-Object-Oriented-Design, PROFESSIONAL-Software-Engineering, SOFT2201-Software-Constr-and-Design-1, SOFT2412-Agile-Software-Development].
- Text area: choose who can commit.
- Text area: repository.

GitHub Organization – Manage Repos. (Revisit)

This screenshot shows the GitHub Enterprise organization page for 'COMP9201'. The top navigation bar includes links for 'Enterprise', 'This organization', 'Search', 'Pull requests', 'Issues', and 'Explore'. Below the header, there's a green square icon representing the organization, followed by the name 'COMP9201'. A main message states 'This organization has no repositories.' with a 'Create a new repository' button. On the left, there are links for 'Repositories', 'People', 'Teams', 'Projects', and 'Settings'. On the right, a 'People' section lists two members: 'bsul6138' (Basem Fathi Suleiman) and 'ffar6831' (Farnaz Farid).

This screenshot shows the GitHub Enterprise organization page for 'COMP9201' after adding repositories. The top navigation bar and organization details are identical to the previous screenshot. The main area now displays four repositories: 'Front-end' (Private), 'Designs' (Private), 'Back-end' (Private), and 'WebStoreApp' (Private). Each repository card provides a brief description and the last update time. The 'People' section on the right shows the same two members: 'bsul6138' and 'ffar6831'.

This screenshot shows the GitHub Enterprise repository settings for 'COMP9201 / WebStoreApp'. The top navigation bar includes 'Enterprise', 'This repository', 'Search', 'Pull requests', 'Issues', and 'Explore'. The repository card shows it's private and has 0 stars and 0 forks. The 'Settings' tab is selected. On the left, a sidebar lists 'Options', 'Collaborators & teams', 'Hooks', 'Integrations & services', 'Deploy keys', and 'Custom tabs'. The 'Teams' section on the right lists three teams: 'FrontEndDeve' (Front-end development team, 2 members, Read permission), 'BackEndDeve' (Back-end Development Team, 2 members, Admin permission), and 'Designers' (Web Application Designers, 1 member, Read permission). A button at the bottom allows adding a new team.

GitHub Organization – Manage People (Revisit)

The screenshot shows the GitHub Organization settings page for 'COMP9201'. The 'People' tab is selected. There are three members listed: Basem Fathi Suleiman (bsul6138), Farnaz Farid (ffar6831), and Hamzah Bin Osop (hos05448). Each member has a profile picture, GitHub handle, 2FA status, privacy level (Private), role (Owner or Member), number of teams (0), and a gear icon for more options. A search bar at the top says 'Find a member...' and there are tabs for 'Members' (selected) and 'Outside collaborators'. A large 'Add member' button is at the bottom right.

- Manage
- Change role...
- Convert to outside collaborator
- Remove from organization

The screenshot shows the GitHub Organization member details page for 'ffar6831' in the 'COMP9201' organization. The member has 3 repositories, 1 team, and is a member of the organization. Their access level is 'Member'. They have 3 repositories: 'COMP9201/WebStoreApp' (Read on this repository), 'COMP9201/Designs' (Write on this repository), and 'COMP9201/Front-end' (Admin on this repository). A search bar at the top says 'Find a repository they have access to...' and there are 'Manage access' buttons for each repository. A gear icon for more options is also present.

GitHub Organization – Manage Teams (Revisit)

The screenshot shows the 'Manage Teams' section of a GitHub organization. At the top, there is a search bar labeled 'Find a team...' and two buttons: 'Import teams' and 'New team'. Below this, there is a table with three rows, each representing a team:

	Visibility	Members	Teams
<input type="checkbox"/> Select all			
<input type="checkbox"/> BackEndDeve Back-end Development Team		2 members	0 teams
<input type="checkbox"/> Designers Web Application Designers		1 member	0 teams
<input type="checkbox"/> FrontEndDeve Front-end development team		2 members	0 teams

You may have 3 repos; Designs, Front-end and Back-end. You want FrontEndDeve to work on the Front-end and Designers team to work on Designs repo and BackEndDeve to work on Back-end repo

The screenshot shows the 'Repositories' section of a GitHub organization. At the top, there is a search bar labeled 'Find a repository...', a button labeled 'Add repository', and four navigation links: 'Discussions', 'Members 2', 'Teams 0', and 'Repositories 2'. Below this, there is a table with two rows, each representing a repository:

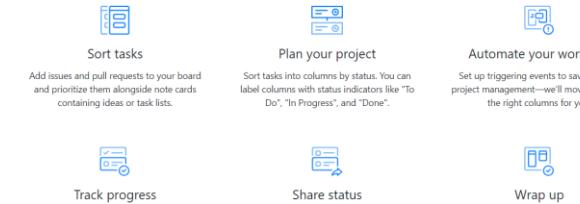
	Last updated	Admin
<input type="checkbox"/> COMP9201/Back-end	updated 21 minutes ago	Admin
<input type="checkbox"/> COMP9201/WebStoreApp	updated an hour ago	Admin

GitHub Organization – Manage Projects (Revisit)



Organize your issues with project boards

Did you know you can manage projects in the same place you keep your code? Set up a project board on GitHub to streamline and automate your workflow.



A screenshot of a GitHub project board titled 'Web App Development'. The board has three columns: 'To do', 'In progress', and 'Done'. The 'To do' column contains a card with 'Enter a note' and a 'Create' button. The 'In progress' column contains cards for 'Create Login Designs' and 'Implement Login', both added by 'bsul6138'. The 'Done' column contains cards for 'Create Sign-up Designs' and 'Implement Front-end Sign-up', also added by 'bsul6138'. A 'Filter cards' search bar is at the top right, and a '+ Add column' button is on the far right. A sidebar on the left provides instructions for creating a new project.

GitHub Organization – Audit Log (Revisit)

- Audit log records all events that have happened at the organization level, who did them and where in the world they were done

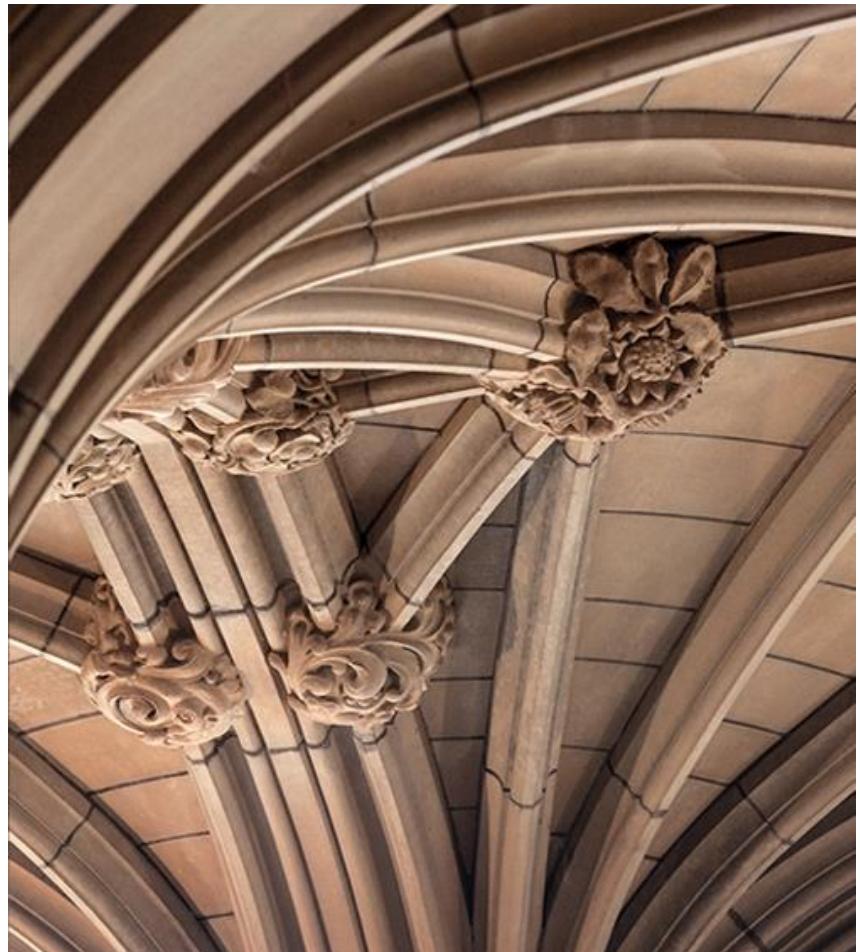
The screenshot shows the GitHub Organization settings page for the 'COMP9201' organization. The left sidebar contains links for Organization settings (Profile, Member privileges, Security, Audit log, Hooks, Installed GitHub Apps, Repository topics, Projects, Teams), Developer settings (OAuth Apps, GitHub Apps), and a 'Recent events' section. The main area is titled 'Audit log' and displays a list of recent events. Each event is shown with a user icon, the event type, the action taken, the repository or team involved, the location, and the time of the event.

Event Type	Action Taken	Target	Location	Time
bsul6138 – project.create	Created project	Web App Development	COMP9201	Australia an hour ago
bsul6138 – team.add_repository	Gave access to	COMP9201/frontendeve	COMP9201/Designs	Australia an hour ago
bsul6138 – team.add_repository	Gave access to	COMP9201/frontendeve	COMP9201/WebStoreApp	Australia an hour ago
bsul6138 – team.add_repository	Gave access to	COMP9201/designers	COMP9201/WebStoreApp	Australia an hour ago
bsul6138 – team.add_repository	Gave access to	COMP9201/backendeve	COMP9201/WebStoreApp	Australia an hour ago

Jenkins



THE UNIVERSITY OF
SYDNEY



Jenkins – CI / CD



- “Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.”
- *Jenkins pipeline* is a suite of plugins which supports implementing and integrating *continuous delivery pipelines* into Jenkins
 - A *continuous delivery pipeline* is an automated expression of your process for getting software from version control right through to end users/customers
 - Typically written in *Jenkinsfile* which is checked in a project’s source code repository

<https://jenkins.io/>

Jenkins – Integration with GitHub (1)

The screenshot shows the Jenkins 'New Item' creation interface. At the top, there's a header bar with the Jenkins logo, a search bar, and user information ('soft2412 | log out'). Below the header, the main form has a title 'Enter an item name' with a field containing 'jenkins-github'. A note below the field says '» Required field'. To the right of the title, there are several items listed with icons:

- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**: Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.

At the bottom of the form, there's a note: 'If you want to create a new item from other existing, you can use this option:' followed by a 'Copy from' button and a 'Type to autocomplete' input field.

Jenkins – Integration with GitHub (2)

Source Code Management

None
 Git

Repositories

Repository URL: <https://github.sydney.edu.au/SOFT2412-Agile-Software-Development/git-local5.git>

Credentials: hos05448/***** [Add](#)

[Advanced...](#) [Add Repository](#)

Branches to build

Branch Specifier (blank for 'any'): */master

[X](#) [Add Branch](#)

Repository browser: (Auto)

Additional Behaviours: [Add](#)

Subversion

Build Triggers

- Trigger builds remotely (e.g., from scripts) [?](#)
- Build after other projects are built [?](#)
- Build periodically [?](#)
- GitHub Branches [?](#)
- GitHub Pull Requests [?](#)
- GitHub hook trigger for GITScm polling [?](#)
- Poll SCM [?](#)

Jenkins – Integration with GitHub (3)

- Webhooks to set up GitHub applications to subscribe to certain events on GitHub
- Events is triggered, HTTP POST payload will be sent to the webhook's configured URL
- Webhooks can be used to update an external issue tracker, trigger CI builds, update a backup mirror

The screenshot shows the GitHub 'Webhooks / Manage webhook' configuration page for the repository 'SOFT2412-Agile-Software-Development / git-local6'. The left sidebar has a 'Hooks' tab selected. The main area shows a form to define a new webhook. The 'Payload URL' field contains 'http://pc-226-2.staff.sydney.edu.au/github-webhook/'. The 'Content type' field is set to 'application/x-www-form-urlencoded'. The 'Secret' field is empty. Below these fields, there are two radio buttons: 'Just the push event.' (selected) and 'Send me everything.'. Under 'Which events would you like to trigger this webhook?', there are two columns of checkboxes. The left column includes 'Commit comments', 'Branch or tag deletion', 'Deployment statuses', 'Wiki', and 'Issues'. The right column includes 'Branch or tag creation', 'Deployments', 'Forks', 'Issue comments', and 'Labels'. Each checkbox has a detailed description below it.

<https://developer.github.com/webhooks/>

Teamwork Collaboration Tools

- What tools would you use to help support your team?
- Examples:
 - Dropbox
 - Google Docs
 - Skype
 - Trello
 - Slack
 - Basecamp
 - Asana
 - ... other

Other Tools – Asana – Project Management

The screenshot shows the Asana web interface for the 'BlueGarden' project. The left sidebar has a dark theme with the Asana logo, 'My Dashboard', 'Invite People', 'Team Conversations', 'Team Calendar', and a 'PROJECTS' section where 'BlueGarden' is selected. The main area displays a task list in 'List' view. At the top of the list is a task titled 'Kickoff:' with a checkbox next to it. Below it is a section titled 'Requirements:' containing two tasks: 'Setup client meeting' and 'Analysis & User Stories'. Under 'Design:', there is one task: 'Technology Stack'. Under 'Development:', there is one task: 'Training'. Under 'Quality Assurance:', there are two tasks: 'Infrastructure' and 'Training'. Under 'Deployment:', there is one task: 'Infrastructure'. Each task has a checkbox to its left and a right-pointing arrow icon to its right.

Task ID	Section	Description	Status
1	Kickoff:	Write project proposal	Completed
2	Requirements:	Setup client meeting	Completed
3	Requirements:	Analysis & User Stories	Pending
4	Design:	Technology Stack	Pending
5	Development:	Training	Pending
6	Quality Assurance:	Infrastructure	Pending
7	Quality Assurance:	Training	Pending
8	Deployment:	Infrastructure	Pending

References

- Andrew Stellman, Margaret C. L. Greene 2014. Learning Agile: Understanding Scrum, XP, Lean and Kanban (1st Edition). O'Reilly, CA, USA.
- S. P. Myers, 2013. [<https://www.teamtechnology.co.uk/team/dynamics/definition/>]
- Cross Dysfunctional Teams.
[<https://mysoftwarequality.wordpress.com/2014/09/04/cross-dysfunctional-teams>]
- Atlassian, Agile Teams. [<https://www.atlassian.com/agile/teams>]
- Tuckman's stages of group development.
https://en.wikipedia.org/wiki/Tuckman%27s_stages_of_group_development
- Atlassian, Teamwork. [<https://www.atlassian.com/teamwork>]
- Team building. [https://en.wikipedia.org/wiki/Team_building]
- Issue tracking system. [https://en.wikipedia.org/wiki/Issue_tracking_system]
- Bug tracking system. [https://en.wikipedia.org/wiki/Bug_tracking_system]

References

- Further Readings:
 - Hackman J. R., “*Leading Teams: Setting the stage for great performances*”, Harvard Business Press 2002
 - Hackman J. R., Katz N. “*Group behavior and performance*”. In Fiske ST, Gilbert DT, Lindzey G *Handbook of social psychology* (5th ed.) New York: Wiley; 2010. pp. 1208-1251. DOI: 10.1002/9780470561119.socpsy002032

Tutorial: Continuous Integration with Jenkins / Docker

Team Dynamics – Team Building Activities

Lecture: Agile Methods, Scrum

