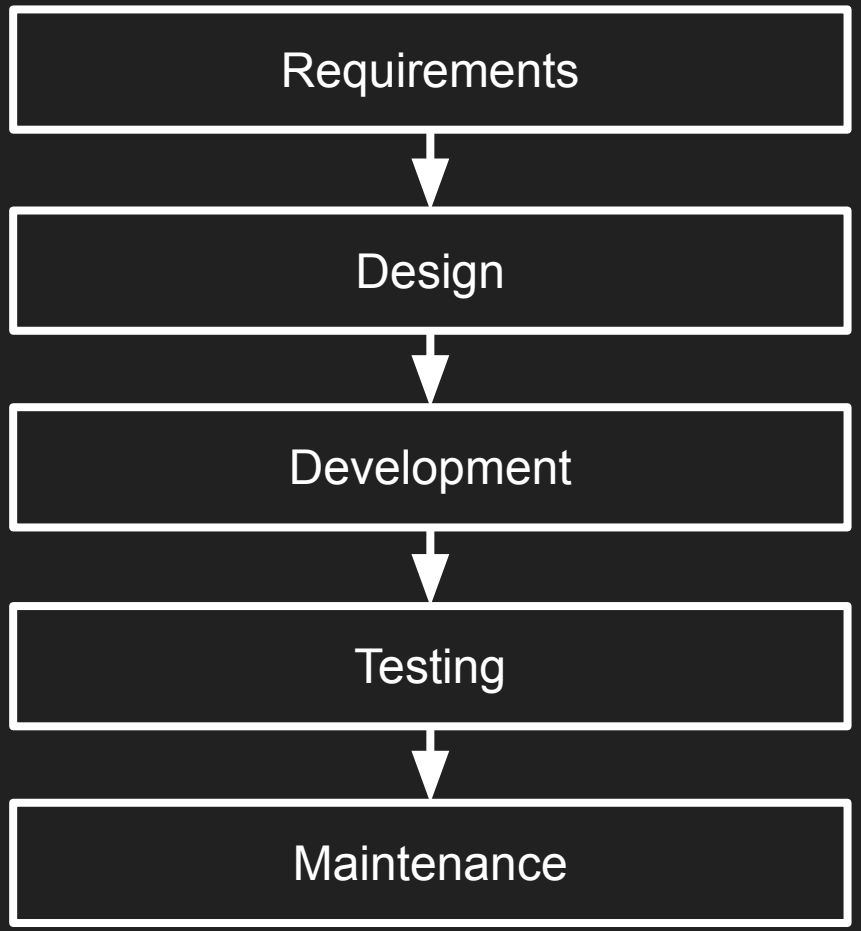
Development Methodologies

Waterfall Method

- One of the oldest methods for software development
- Complete each step before moving on to the next, rarely revisiting previous stages
- Documentation driven, specifications should be created before any development
- Good for large and critical applications, especially safety or compliance driving systems



Waterfall Pros & Cons

+ Software process subject to discipline, planning, and management
+ Postpone implementation to after understanding objectives
+ Time spent upfront can save time later during development
- No feedback between steps, and steps are rarely revisited
- No development parallelism, lock-step development between steps
- Single monolithic delivery date, integrations often happens near the end
- No changes anticipated, build everything correctly the first time
- Important stages (testing) reserved to the end when time is tightest

Waterfall Use Cases

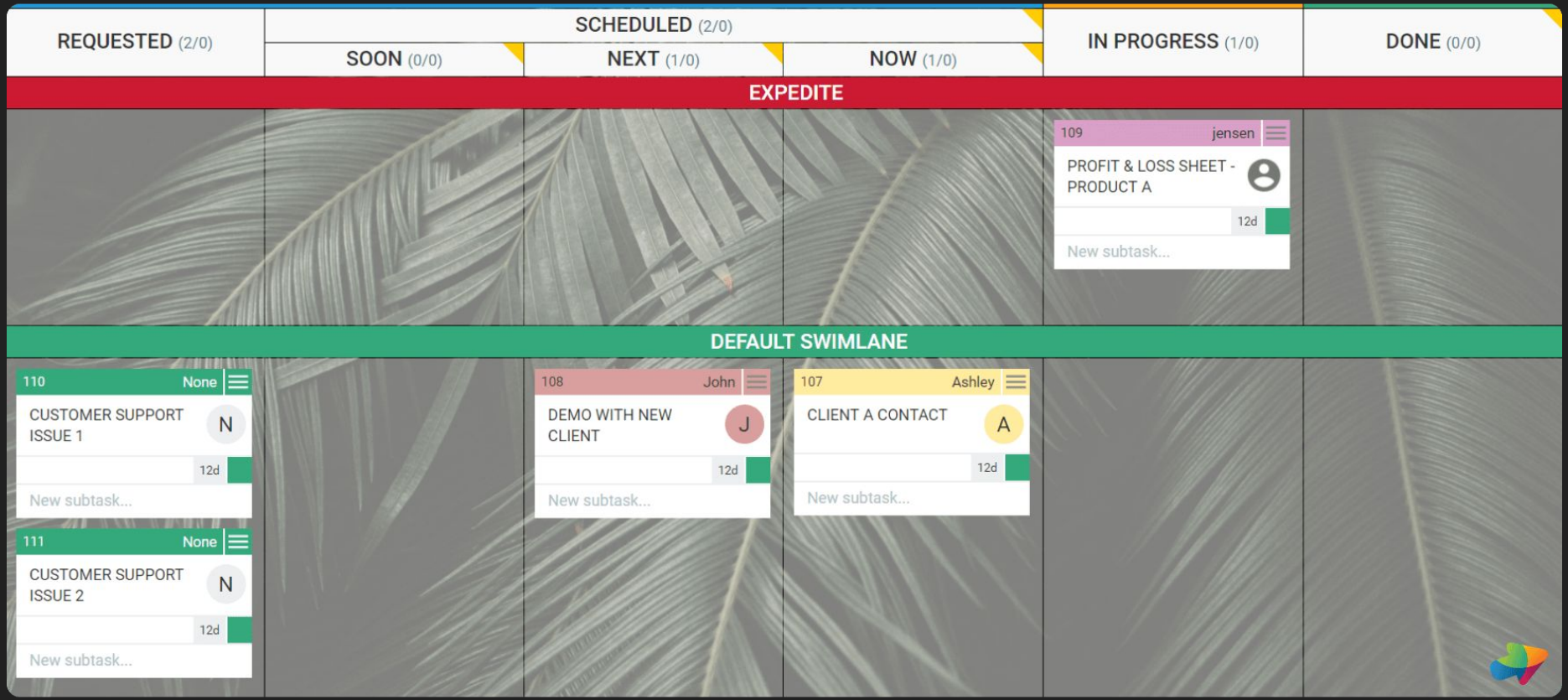
- Safety Critical Systems
  - Systems where there can be grave consequences if the software does not operate properly at all times. Aviation, military, and medical systems are often examples of this type of system.
- Legally Mandated Systems
  - Systems where a large part of their requirements are specified by a law, required specifications, or government agency. Health Insurance Portability and Accountability Act (HIPPA) compliance for medical records is an example of this type of system.
- Contracted Systems
  - Systems where the specifications of the project and their completion are defined in the contract. These are typically third-party development projects where an end-product is delivered by a set date.

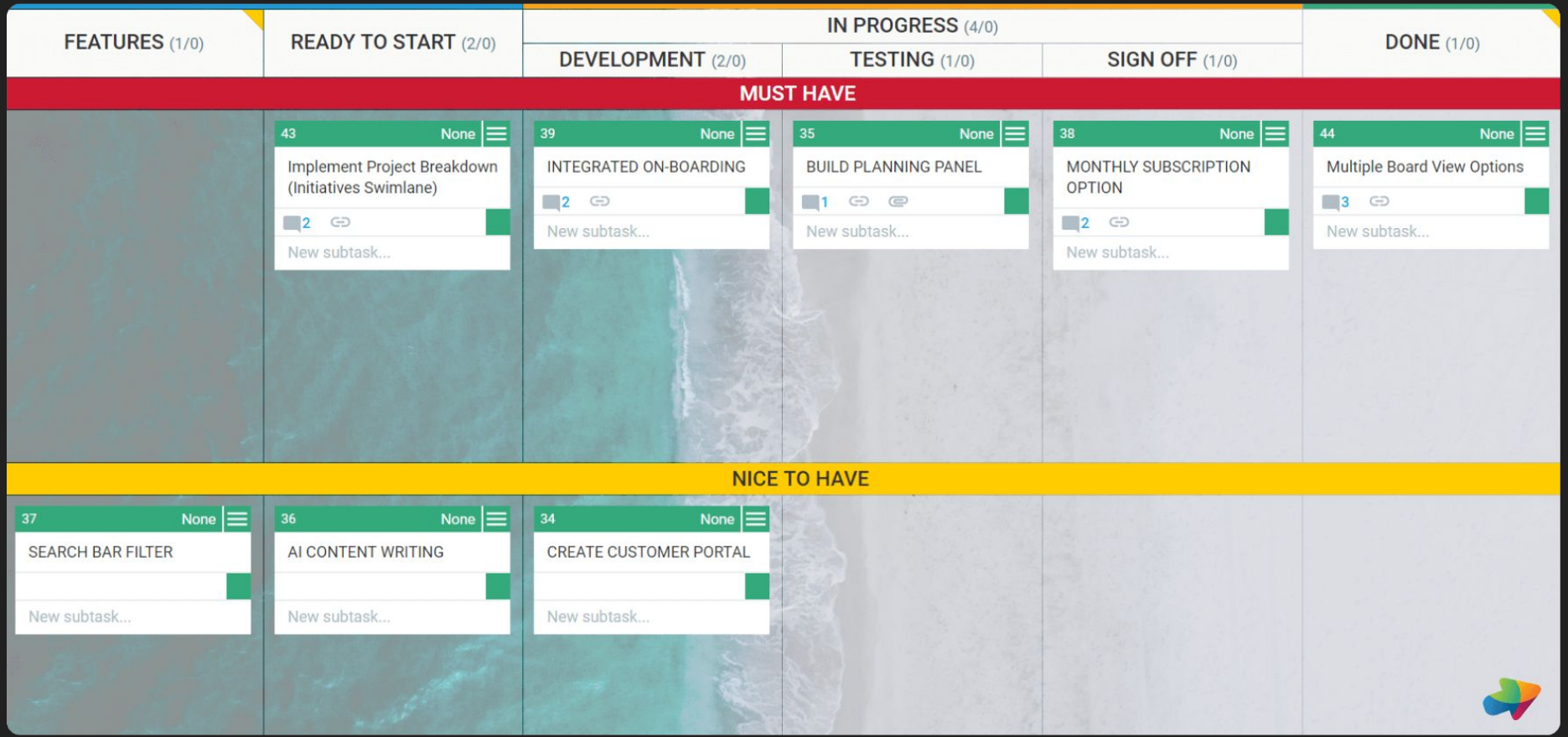
Agile

- Set of principles/values for software development (and other things)
  - 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 isn’t an individual method, but a philosophy which is used in lots of different methods
  - **Kanban**
  - **Scrum**
  - Etc.

Kanban (Signboard)

- Lean/Just-in-time manufacturing method developed by Taiichi Ohno at Toyota
- “Limiting the number of pending requests makes the process more sensitive and reveals inefficiencies.”
- This process focuses on creating bins/columns/stages that represent the pipeline a task needs to advance through to be completed
- The number of tasks that can exist in each part of the pipeline is then limited to increase throughput
- This is usually illustrated through tasks represented as cards, which are moved along different columns on a board to represent their position in the pipeline


REQUESTED (2/0)
SCHEDULED (2/0)
SOON (0/0)
NEXT (1/0)
NOW (1/0)
IN PROGRESS (1/0)
DONE (0/0)
EXPEDITE
109 jensen
PROFIT & LOSS SHEET - PRODUCT A
12d
New subtask...
DEFAULT SWIMLANE
110 None
CUSTOMER SUPPORT ISSUE 1
N
12d
New subtask...
111 None
CUSTOMER SUPPORT ISSUE 2
N
12d
New subtask...
108 John
DEMO WITH NEW CLIENT
J
12d
New subtask...
107 Ashley
CLIENT A CONTACT
A
12d
New subtask...


FEATURES (1/0)
READY TO START (2/0)
IN PROGRESS (4/0)
DEVELOPMENT (2/0)
TESTING (1/0)
SIGN OFF (1/0)
DONE (1/0)
MUST HAVE
43 None
Implement Project Breakdown (Initiatives Swimlane)
2
New subtask...
39 None
INTEGRATED ON-BOARDING
2
New subtask...
35 None
BUILD PLANNING PANEL
1
New subtask...
38 None
MONTHLY SUBSCRIPTION OPTION
2
New subtask...
44 None
Multiple Board View Options
3
New subtask...
NICE TO HAVE
37 None
SEARCH BAR FILTER
New subtask...
36 None
AI CONTENT WRITING
New subtask...
34 None
CREATE CUSTOMER PORTAL
New subtask...

RoseHack 2019 @ UCR

Kanban Pros & Cons

+ Tasks are not prescribed a certain size or deadline, and can all travel through the system independently
+ Works well for both individuals and small teams, with specialized and generalized contributors
+ Allows for large variety in the types of tasks which can exist on the board
- Lack of timeframes or priorities is not always realistic (priority and lanes were developed to help solve these issues)
- Process can become chaotic with larger teams, where having a lead can be beneficial

Kanban Use Cases

- Individual Development/Organization
  - Creating a Kanban board for your individual development projects or even for your normal activities. This method can help you stay focused and stop you from jumping between many tasks without making major progress on any of them
- Small Group Development
  - When working with a small group of developers who all have equal ownership of the project the first column becomes a global task queue and you can limit your work based on the number of developers working on the project
- Organization between Distinct Groups
  - When working with distinct groups within an organization that have well defined ownerships of different development stages, this method can be useful for identifying and addressing bottlenecks and stopping groups from being overwhelmed. For example, developers can only push so many tasks to testing before their queue is full.

Scrum

- Lightweight agile process for small, cross-functional, self organized teams
- Has a small number of roles to keep members focused and make sure neither business or engineering overtake the development conversation
- Scrum Master
  - Typically filled by project manager or team lead
  - Main job is to remove impediments from scrum team members, enforce scrum principles
- Scrum Team
  - Typically 5-10 cross functional (programming, testing, design, etc.) members
  - Members should be full time and can only change teams between sprints
- Product Owner
  - Acts like a single voice for the “user”
  - Knowns what features (not development tasks) need to be build and in what sequence
  - Typically a product manager

Sprint Cycle

1. Product Owners create a product backlog of features that need to be developed, and prioritize that backlog based on current needs
2. The highest priority tasks are broken down into small deliverable tasks, which are then prioritized by developers and estimated for time
3. Based on number of team members and how many hours each team member can contribute, an equivalent number of tasks are assigned to members that can be completed in the next **sprint**, a short period of time where development goals are fixed and cannot be changed (usually 2 weeks)
4. Information and issues are shared between team members through ****daily stand-up meetings****
5. After each sprint, goals and progress are re-assessed and re-prioritized and a new sprint is planned

Spring Planning Example

- Spring Cycle: 2 weeks
- Total work hours per day: 8 hours/day
- Other work hours per day: 2 hours/day
- Actual work hours per day: 8 - 2 = 6 hours/day
- 6 hours/day * 5 days/week * 2 weeks = 60 hours/sprint
- Each team member can support 60 hours worth of work in a sprint
  - There are lots of other ways to do these types of estimates, but hours is common
- Scrum Team Size: 5 members
- 5 members * 60 hours/sprint = 300 hours/sprint total team effort
  - Not every member will have the same other work hours, so this is usually a sum not multiple

Typical Standup Meeting (5-10 min)

Planking Standup Meeting (1-2 min)

Scrum Pros & Cons

+ Provides balance between developer interests and business interests
+ Provides dedicated roles, allowing team members to concentrate on developing
+ Allows business priorities to shift without creating a moving development target
- Creates organizational overhead (sprint planning meetings, standup meetings, review meetings, user story meetings, etc.)
- Allows for feature creep since there is no defined end-date or final product
- Losing team members can create large negative impacts on the team/project

Scrum Use Cases

- Scrum is probably the most used method currently in industry
- Many companies/teams use either all or some of the scrum process
  - Fully Scrum: dedicated members with roles and dedicated meeting times
  - Medium Scrum: disconnect between product owner and scrum team, manager de-facto scrum master (this often happens in internal teams where developers can “be their own product owners”)
  - Light scrum: daily standup meetings