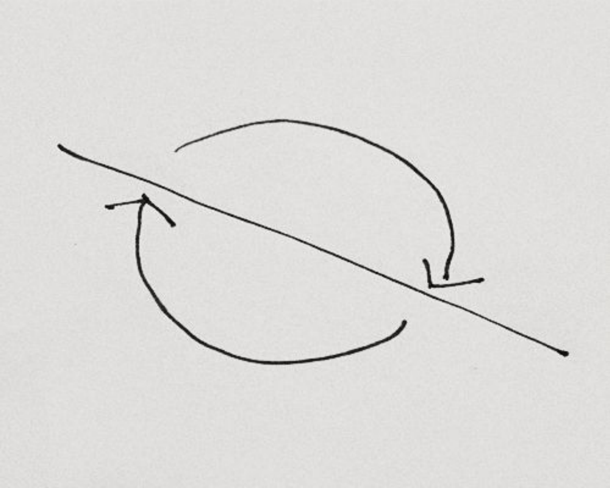
Value-Driven Development with “Continuous Discovery”



Hello

Nice to meet you

Introductions

Prabhat Sinha

Vishnu Vijayan Padmanabhan

Jordan Ryan

Agenda

Why Discovery?

Discovery Workshops

Shortcomings of Discovery Workshops

Agile discovery

Best practices

Evolving Agile with Continuous Discovery

Problem Statement

How do you manage...

- Imperfect understanding of scope
- To maximize the value delivered to the client
- Changes in scope due to internal, external factors

Why Discovery?

Why Discovery?

Waterfall Discovery

- Requirements Engineering
- SRS
- Gantt Charts
  - Lots of Gantt Charts
- Use Cases

Agile

- User Stories
- Backlog Grooming
- Use Case 2.0
- Vertical Slicing

Example of Non-discovery project

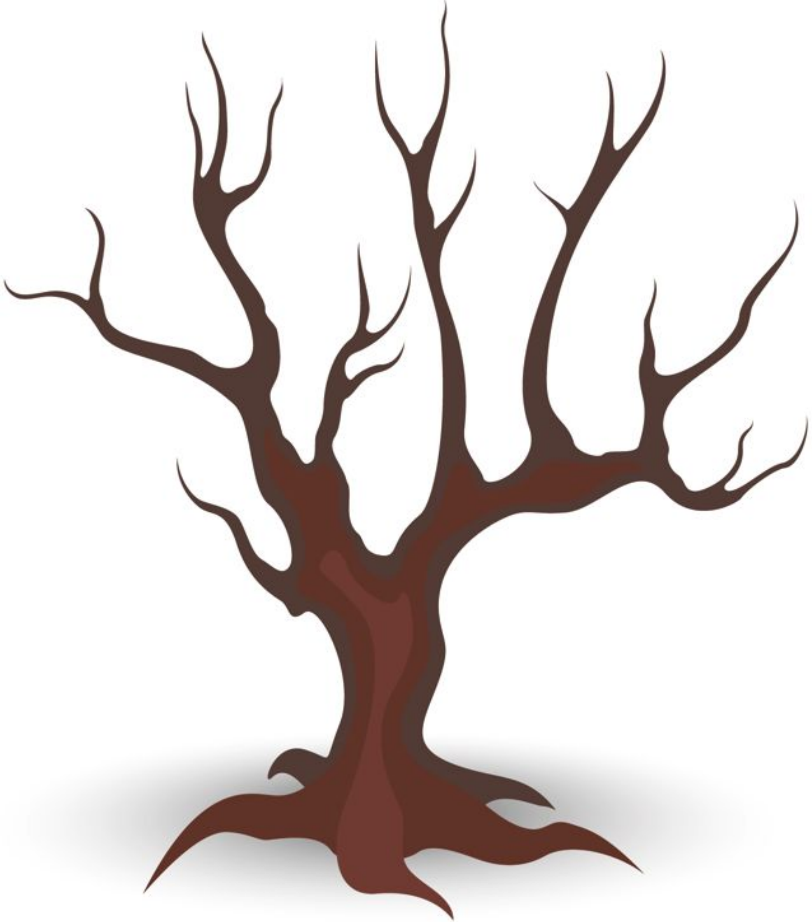
E-learning Application

- Synopsis - The project was intended to provide equal learning opportunities to students from all categories.

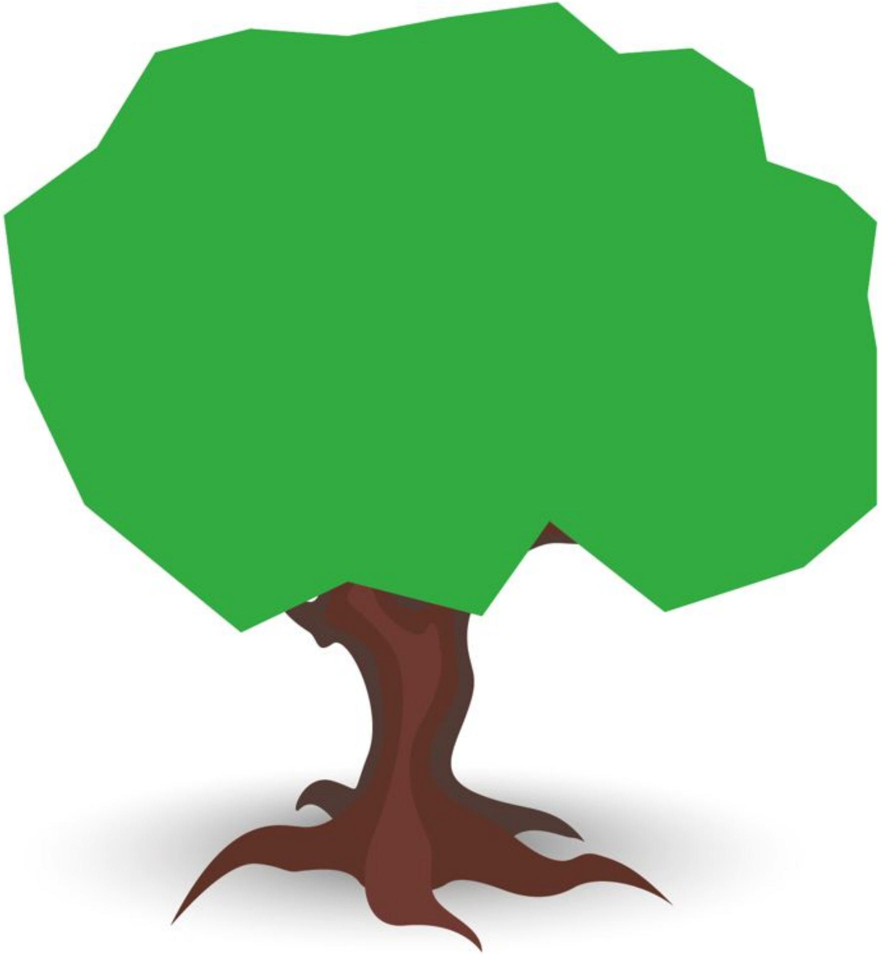
- Problems after project completion
  - Content Quality Issue
  - Security issues for platform
  - User response dropped 95% during the initial pilot itself.

Example of Non-discovery project……

- **Reason of failure -**
  1. Missing requirements
  2. Couldn’t gain stakeholder support
  3. No User Research
  4. Quality related issues
  5. Security related issues
  6. Couldn’t recognize the transformational shift for the users



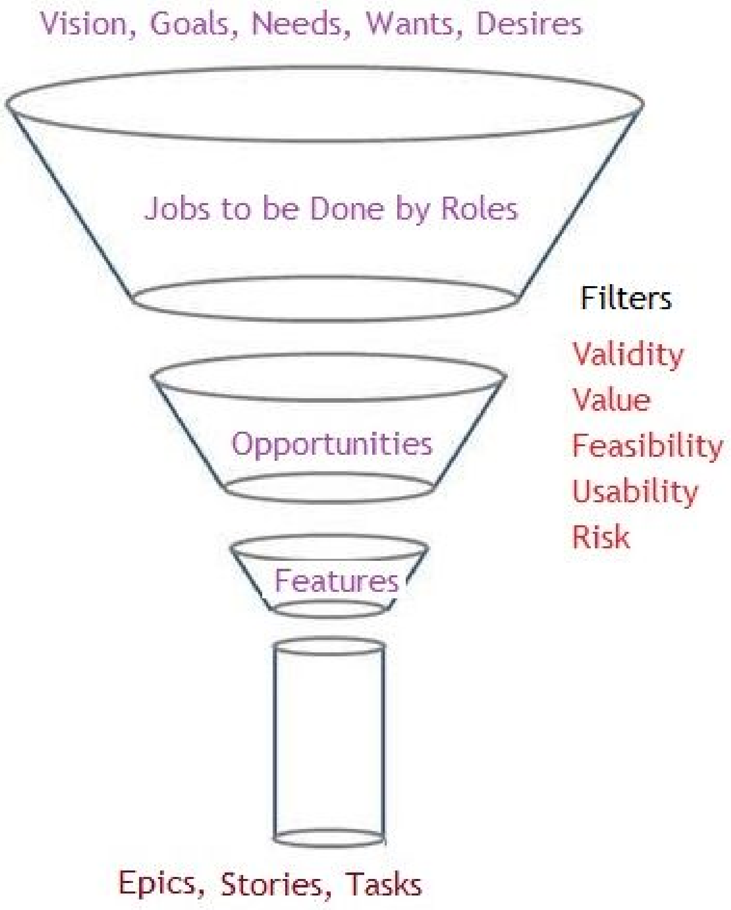
The Feature Tree



Feature Tree After Discovery Workshop

Narrow the Cone of Uncertainty

- Do we have the skillset?
- Do we have the same understanding as the client?
- What are client's expectations in terms of quality and cost?
- What kind of risk are we taking on with this client in terms of their temperament or expectations?
- What is uncovered in Technical Discovery?

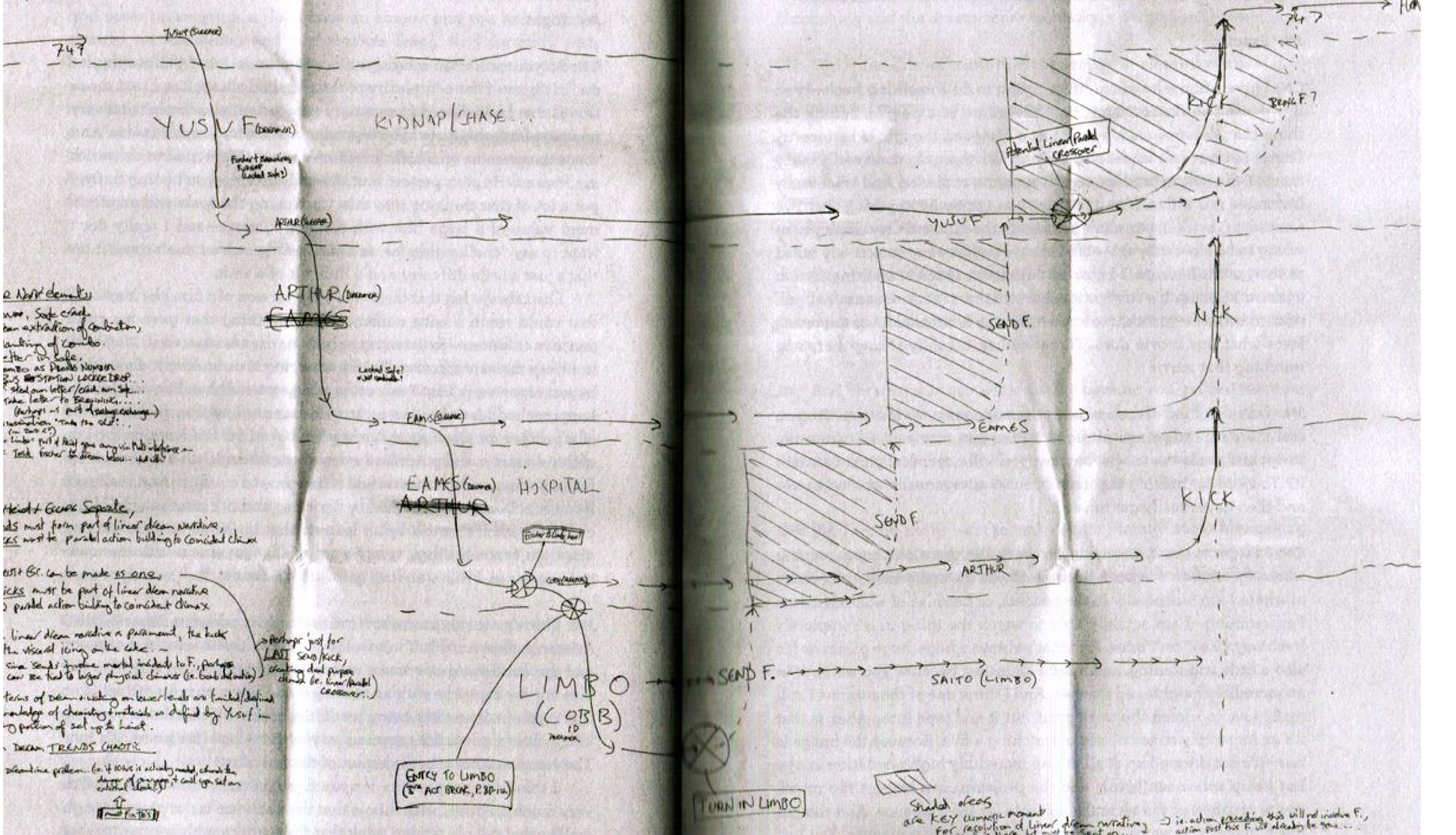


Answering these Questions.....

… is the first step ***before*** Agile starts. We ***measure the risk*** of each project with a Discovery Workshop.

However…

Discovery should not stop there.



e Nort elements

[illegible], Safe crack.
[illegible] extraction of combination,
[illegible] planting of combo.
[illegible] letter in safe.
[illegible] combo as Phone Number
[illegible] STATION LOCKER DROP...
[illegible] steal own letter/crack own safe...
Take letter to Browning...
(perhaps as part of [illegible] exchange)
[illegible] "Take the Safe"...
(in [illegible])
[illegible] "limbo" part of [illegible] in [illegible] [illegible]
[illegible] Task Fischer to dream [illegible]... [illegible]

Heist + Escape Separate,
[illegible]ds must form part of linear dream narrative,
[illegible]cks must be parallel action building to coincident climax

[illegible] + Esc. can be made as one,
[illegible]icks must be part of linear dream narrative
[illegible] parallel action building to coincident climax

[illegible] linear dream narrative is paramount, the kicks
[illegible] the visceral icing on the cake. –
[illegible] since 'Sends' involve mortal incidents to F., perhaps
[illegible] can be tied to larger physical climaxes (i.e. bomb detonation)

→ Perhaps just for LAST send/kick, creating dual purpose climax (i.e. linear/parallel crossover...)

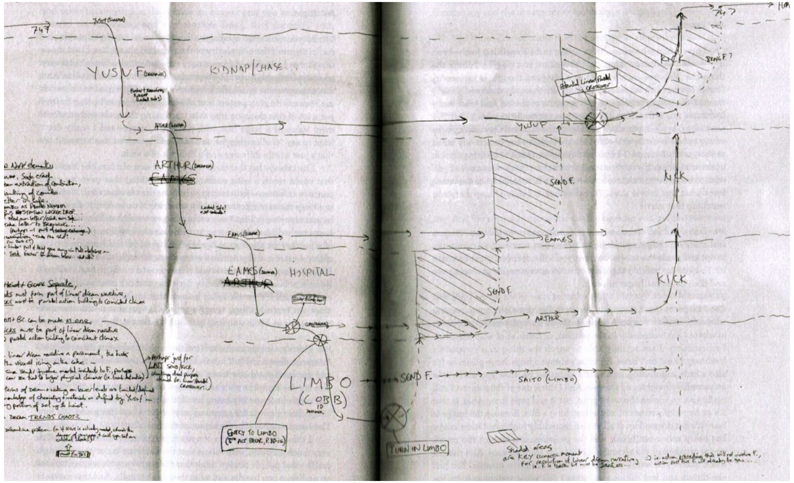
[illegible] terms of Dream-sinking on lower levels are limited/defined
[illegible] knowledge of chemistry + materials as defined by Yusuf in
[illegible] portion of set up to heist.

[illegible] Dream TRENDS CHAOTIC

[illegible] problem. (i.e. if [illegible] is actually [illegible], [illegible] [illegible] + [illegible] set an [illegible])

[illegible]

Shaded areas are KEY climactic moments for resolution of linear dream narrative [i.e. F. is here but must be sent so...] ⇒ i.e. action preceding this will not involve F., action post this F. will already be gone...



Inception

Long term discovery is important to deliver a product of quality.

Discovery Workshops are about Mitigating Risk

Mitigating risk for the agency

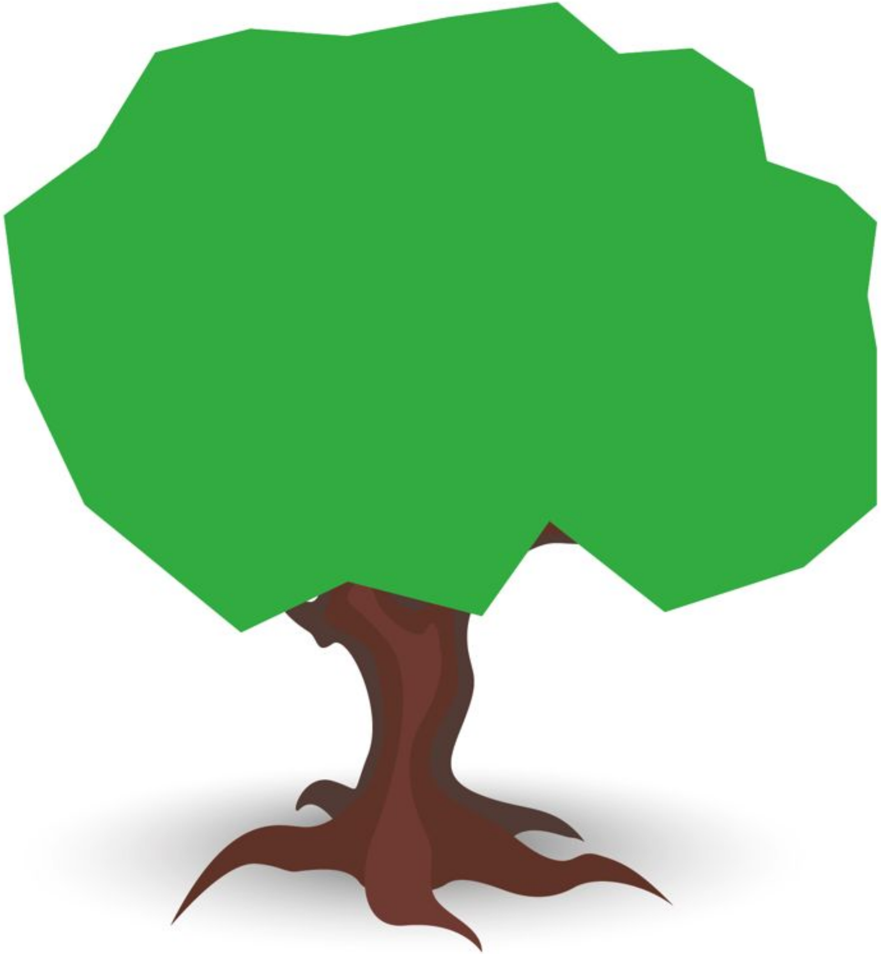
- Price the project for the amount of resources you’ll need to execute properly
- Set up your team to excel (and never fail)
- Align the team to the project
- Engage solutions architects and business analysts who will understand and guide the development team
- Avoid hidden costs of undocumented logic
- Get buy in for solutions from key stakeholders

Mitigate risk for the development team by

- better understanding scope
- better understanding old technology
- better understanding external factors on timeline (deadlines, external/internal events)
- better understanding underlying goals for the business
- better understanding defining metrics or analytics for project success

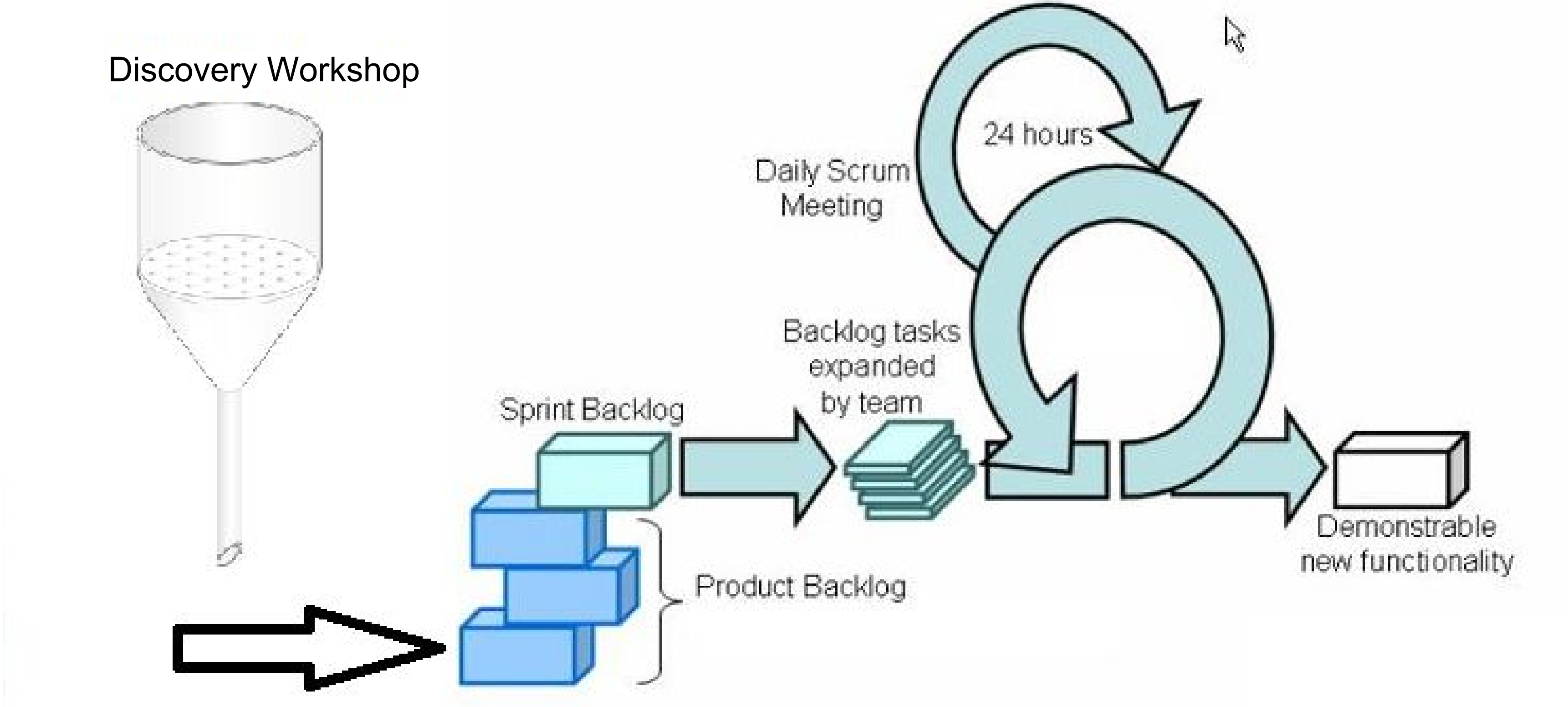
Mitigate risk for the client by

- understanding budget vs. implementation
- demonstrating trade-offs between quality and timeline
- verifying assumptions made by solutions architecture
- executing POCs to verify understanding and shared vision
- gives opportunity to articulate needs, desires, vision



Feature Tree After Discovery Workshop

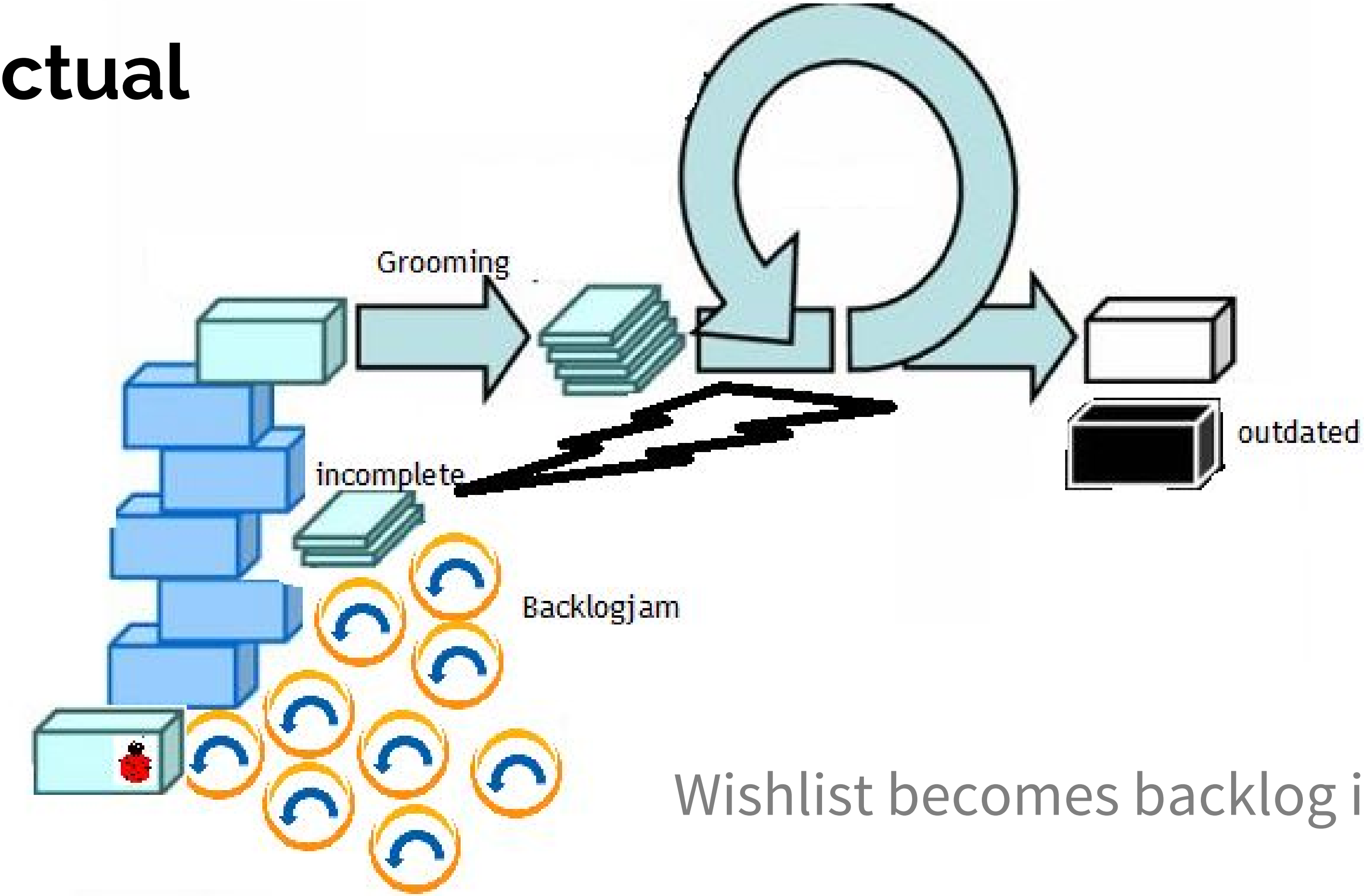
Discovery Workshop + Agile development



Discovery Workshop

24 hours
Daily Scrum Meeting
Backlog tasks expanded by team
Sprint Backlog
Product Backlog
Demonstrable new functionality

Actual



Wishlist becomes backlog items !

But Risk is only mitigated 50% by planning.

Feature Tree Without Continuous Discovery

Discovery is about Learning

You can mitigate the other 50% of risk with continuous learning.

Feature Tree During Continuous Discovery - Sprint 2

Discovery workshop is best for

- Product / Problem definition, establishing business goals
- End user research, identifying solution opportunities
- Analysis & building epic backlog
- ball park estimation of effort

But you can only transfer this knowledge to the team in slices, or time boxed sprints.

You can’t plan for everything.

You can’t prepare for the unknowns

Waterworld spent

$100 million in

pre-production, but…

a hurricane during

shoot almost doubled

the cost

Example of Discovery Blindspots

MMORPG

> 1-year+

> Business Market Changes

> Development on Old Platform continues, must manage delta

> Feature Details extremely nuanced

> Evolving Design Requirements

So…
how do we make
Discovery Continuous?

One sprint at a time.

Evolve Agile

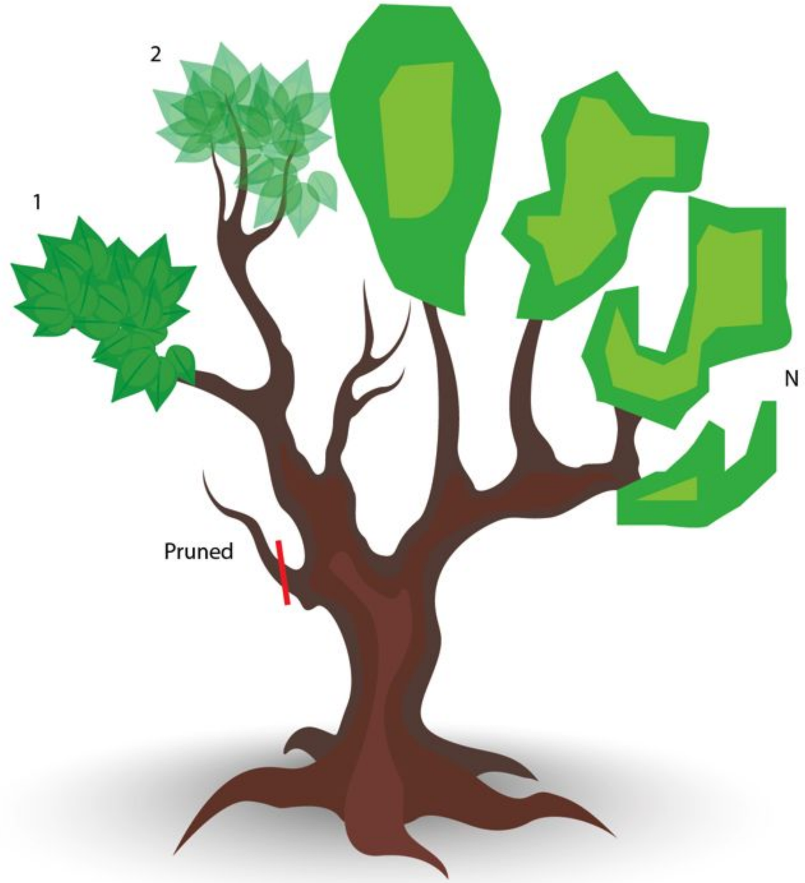
With Continuous Discovery

- We scope stories, requirements, & value proposition each sprint
- Team members learn on each iteration
- Higher-resolution of understanding is achieved with Agile Discovery

Feature Tree During Continuous Discovery - Sprint 1

If your development team
is discovering requirements
each sprint…

You should be defining
features each sprint.



Stories Nurtured on each prior Sprint

Agile & Continuous Discovery

CRs, Bugs, and New Scope Need to be Prioritized

Use constant learning to deliver value

The original script only said, "Travis talks to himself in the mirror." De Niro pretty

much improvised everything

as he learnt more and more

about the character

while playing the role

How does Continuous Discovery work?

Step 1

Capture a Value Statement.

“We want Announcement blocks to give timely updates of contests, sweepstakes, to drive X% of engagement for that content offering.”

Step 2

Verify requirements.

User Story
& Acceptance Criteria
Validation

Step 3

Prioritize & Prune

Bugs
Change Requests
Epics
Stories
Tasks

All need sprint-by-sprint prioritization.

Step 4

Validate understanding.

Review value statements with client & team.

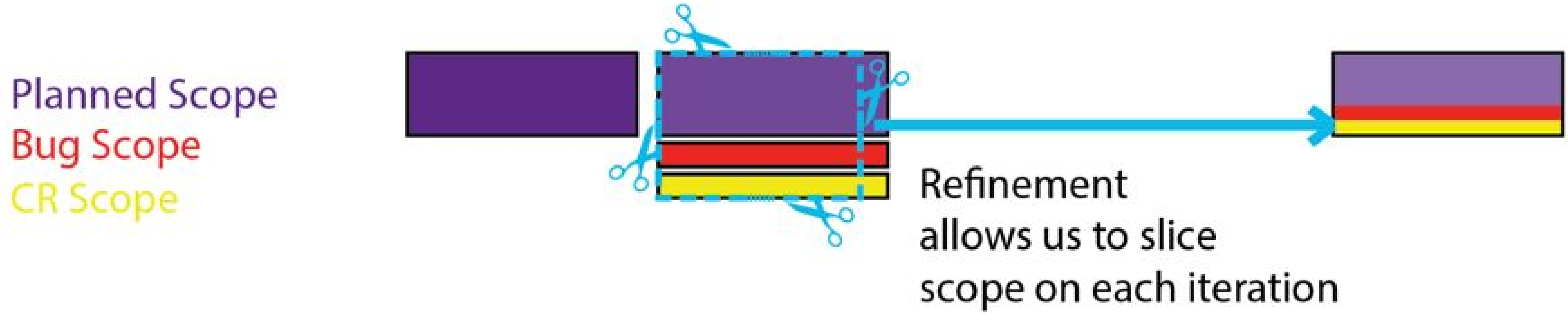
Step 5

Execute.

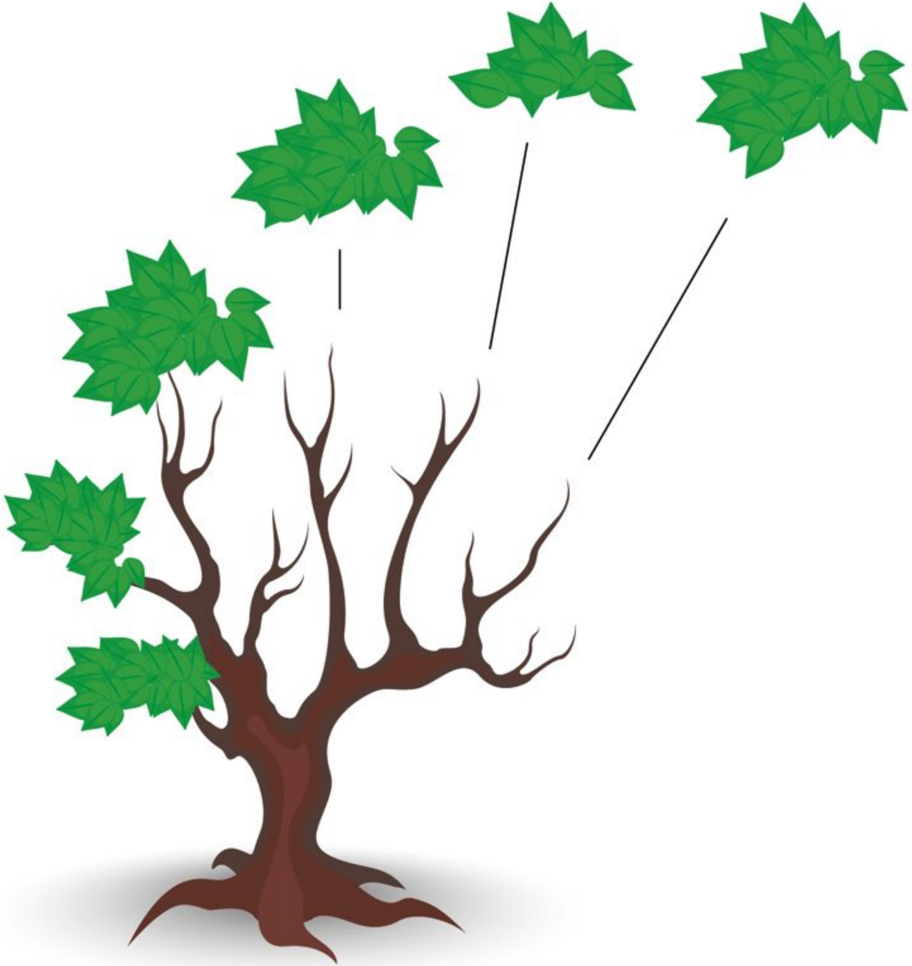
Accelerate execution!

Verify and Refine Scope

Backlog grooming = {discovery validated CRs} + {discovery validated product backlog} + {bugs}



Refinement
allows us to slice
scope on each iteration



Feature Tree During Continuous Discovery - Sprint N

Continuous Discovery is Backlog Verification.

Example of Agile Discovery

Legacy.com

- 6 week discovery
  - Evolving requirements per Node/React
  - Fast paced delivery timeline, < 6 months
  - Manage requirements delta across 2 teams
  - UAT, Migration, Scope Changes
  - Challenges with Decoupled Drupal, Caching
    - must continually re-prioritize, re-slice, re-scope and time box the sprint

Best practices of continuous discovery

1. Verify scope 2 sprints ahead of delivery.
2. Keep all stakeholders on the same page.
3. Understand ecosystem of product.
4. Continuous communication with end users to validate assumptions.
5. Properly define user needs in product features.
6. Keep adding details to product features so that Product Backlog is ready for development.
7. Record updated requirements as the company’s needs or marketplace changes. Track requirements delta.

current		continuous
1. Long time between defining epics and execution	→	1. Shorter time between epic definition and execution
2. Groom epics, stories	→	2. Validate assumptions, prune features
3. Accommodate changes	→	3. Accept business value & market changes
4. Continuous development	→	4. Deliver best value within budget

Benefits

- Ensure that you **always develop the most valuable thing, first.**
- Development can start early, speed of execution is improved
- Agency discovery teams gain opportunity to understand the stories as they evolve.
- Data-driven development instead of data-informed development.
- Minimize stories developed for outdated requirements.
- Deliveries are continuously aligned to business values.
- Small delta between time requirements defined and developers review reqs.

Plan for the unexpected, by not having a plan until the week before.

Closing Thoughts

- Drupal projects greatly benefit from continuous discovery.
- Initial discovery sets high level business goals - Drupal gets 90% of work done with initial site installation.
- Continuous discovery gets the finer details, confirms overall direction - Last 10% of drupal website is achieved through customisation & fine tuning.

Questions?
Follow Us

we’re friendly.

Prabhat - @prabhatfun
prabhat@axelerant.com

Vishnu - @pvishnuvijayan
vishnu@axelerant.com

Jordan - @jordan_ryan
jordan@facetinteractive.com