

# My Experience Scaling

*How we scaled Agile to meet our needs.*

David Hanson

May 2020

# The Way Forward



David Hanson

## About the Presentation

One of Agile Boston 2020's goals is to share success stories with the local Agile community.

This presentation outlines how we scaled Agile to meet the needs of a multi-million dollar multi-year program to create a common platform for portfolio management.

Our "small-scale" scaled Agile implementation most closely resembles LeSS.

The scaled implementation, initiated in 2014, has been largely successful, but not without some compromise, and continues today with remaining opportunities for improvement.

## About the Presenter

David has extensive experience implementing numerous Agile models, scaling when needed, and transforming organizations.

Since 1999, David has experimented with XP on an internal Lean Startup, transformed a RUP project by implementing continuous flow, and transformed a traditional waterfall organization by first introducing Kanban, then Scrum, forming our own interpretation of Scrumban.

David's earliest scaled implementation in 2008 successfully leveraged a plan-build-operate model. David's next scaled implementation was branded "The Way Forward" and is covered here.

# Background

## My approach to Agile

- Kanban > Scrum > XP
- Start small
- Experiment and evolve
- Make it work
- *Then study*

## Similar approach with scaling

## Before scaling

- Bangalore team following RUP
- Mumbai team following Spiral
- Transitioned both teams to Scrum
- Teams small (-ish), stable, cross-functional

# Need to Scale

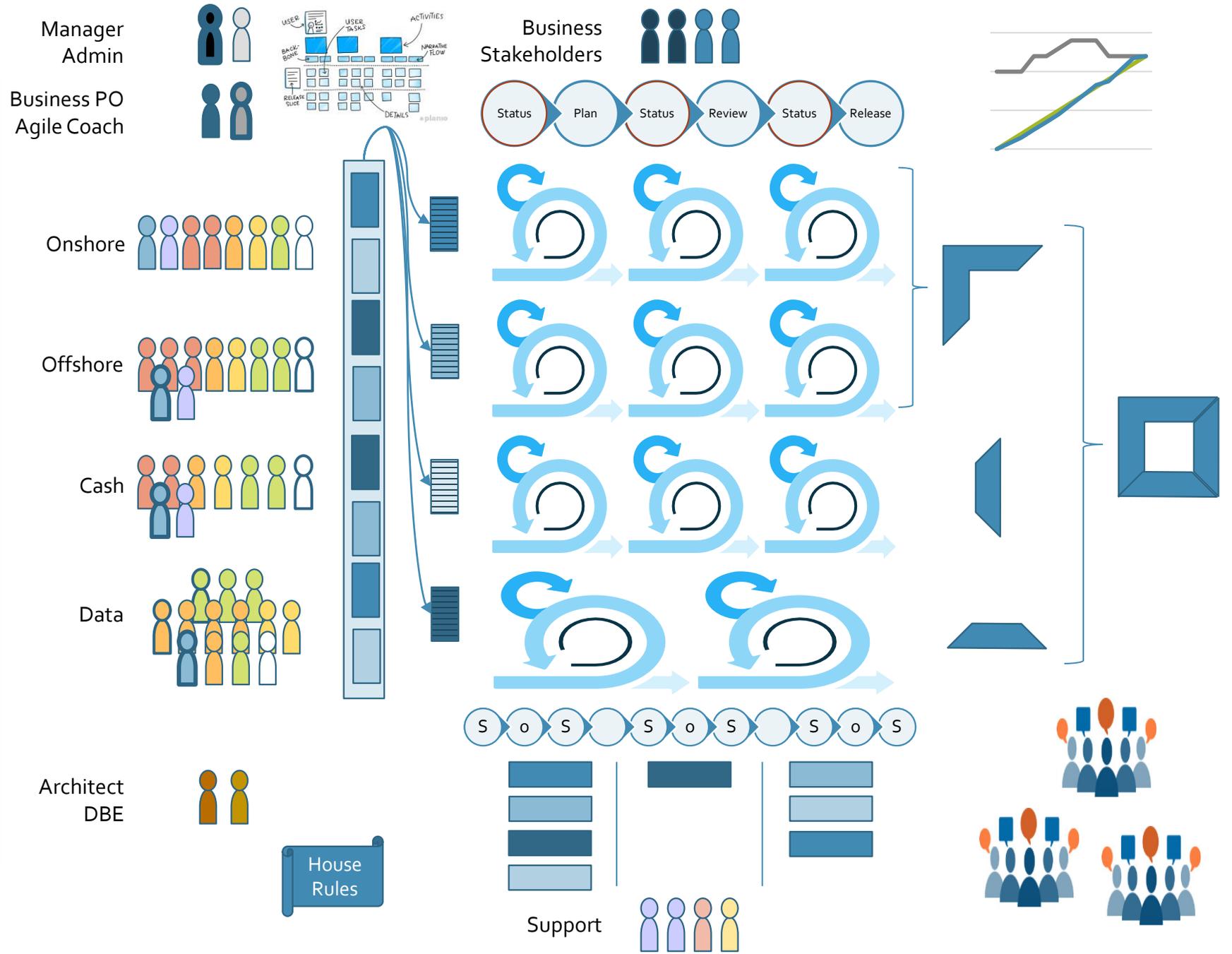
- Teams had been supporting independent projects for three years
- New program initiated to create one common platform for all Fixed Income and Cash portfolio management needs
- Inherited another large team practicing *managed* Scrum

# Breakout

Did you have a need to scale?

- What's the largest team you've been on?
- What's the largest team of teams you've been part of?
- If you were on a large team, might you have benefited from scaling?
- If you were on a team of teams, what scaling model did you use?

# Our Small Scaled Scrum



# Management Roles

## Governance Structure:

- Business and development managers for strategic decisions
- Business stakeholders for tactical decisions
- Business sponsored a lead product owner
- Development manager acting as Agile coach

## Challenges:

- Boston and London offices had different strategic and tactical goals
- Business nominated 6 stakeholders
- Product owner not empowered
- My development manager drove product vision

# Focused Teams

- Split inherited team into Onshore team and Offshore team
- Focused Onshore on core development
- Focused Offshore on legacy and support
- Focused Mumbai team on subset of program
- Focused Bangalore team on common data layer

## Challenges:

- Management overrode proposal to create full vertical teams
- Common data layer team was large and continued many traditional practices
- Data team transitioned to report to different manager
- Inherited offshore team later split between support and cash

# Scaling Model Roles

*a little more than LeSS*

- Program product owner and proxy product owners for each team
- Program Agile coach and Scrum master for each team
- Development teams all had analyst, developers (UI, MT, DB), and testers
- Cross-team staff included a dedicated architect and DBE
- Offshore teams each had proxy product owner and analyst both onshore

## Challenges:

- Proxy product owners more influential than program product owner
- Team members reported to Scrum masters reported to product owners
- Large data team formed unofficial development and QA sub-teams

# Scaling Model Events

*LeSS might have been  
more*

- Synchronized sprints with releases every 6 weeks
- Scrum of Scrums: obstacles, dependencies, risks, focus; speaker and observer
- Scaled Planning: mid-sprint between product owners, team representatives, and managers
- Scaled Review: mid-sprint between product owners, team analysts, and business stakeholders
- Teams maintained independent Scrum events, with *staggered* meeting times
- Weekly management meeting with project radiator, impediment backlog, backlog burnup, and adoption metrics

## Challenges:

- One team declined to shift from 3 week to 2 week sprint
- Management rule that meetings between business and technology should have equal representation
- Business stakeholders lacked patience for detailed review of stories in Sprint Review

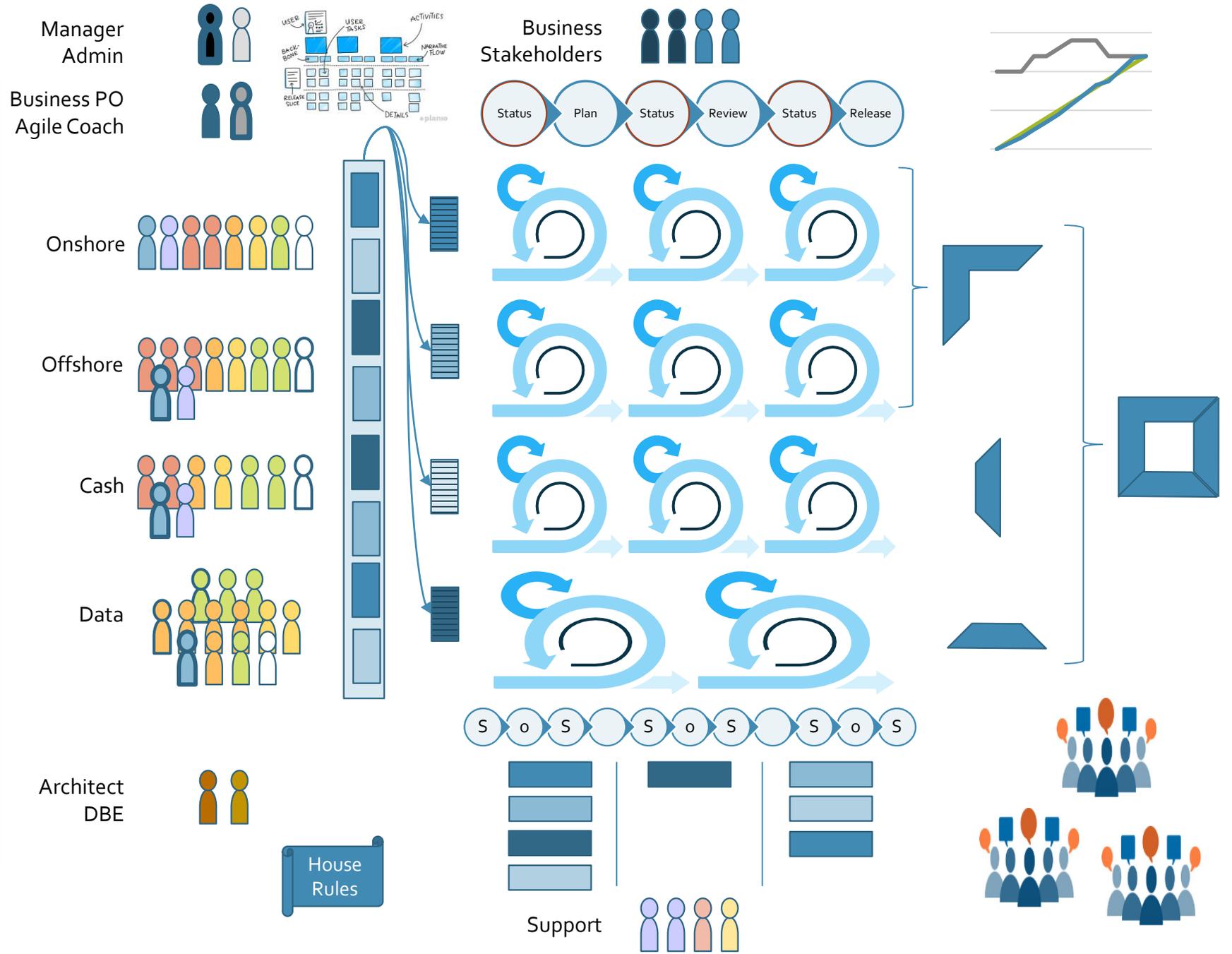
# Scaling Model Artifacts, Agreements, Communities

- Started with epic story map > single product backlog
- Sprint backlogs continued by team
- Potentially shippable increment
  - Integration sprint > integration every sprint > code integrated daily
  - Defined potentially shippable as ready for UAT
- House Rules included minimum standards for sprint events and definitions of ready & done

## Communities of Practice:

- Cross-team (long-standing knowledge sharing)
- Scrum Master (retro of retros)
- Development (integration, deployments)
- QA (regression, automation)
- DevOps (company-wide)

# Our Small Scaled Scrum



# Success and Compromise

## Success:

- Cross-team SMEs: dedicated to team of teams leveraging Kanban
- Common Support: no interrupt buffer
- Scrum of Scrums: format, coverage, attendance
- Integration Sprints (teams self-organized to daily)
- House Rules: effectively our Scrum Guide
- Knowledge Sharing: continuous learning culture

## Mixed:

- Proxy Product Owners (lacked independent strategic PO)

- Staggered Events: allowed ambassadors from other teams and interested managers to attend
- Story Mapping (extend to broader audience)

## Compromise:

- Synchronized Sprints (w/o, difficult to reduce release cycle)
- Vertical interface with horizontal database (UI teams avoided data layer)

## Opportunity:

- Replace weekly management with EAT and scaled planning and review with EMS

# Observations

- The Scrum Framework alone provides a good framework to scale
  - Consider Roles, Events, Artifacts, plus Agreements, Metrics
- Scaling provides opportunities which may not exist with one team
  - Shared SMEs dedicated to team of teams
  - Support team as alternative to interrupt buffer
- Scaled Agile provides a natural opportunity to create learning communities and accelerate adoption
- Successful scaling doesn't necessarily require SAFe or S@S

# Our Small Scaled Scrum

