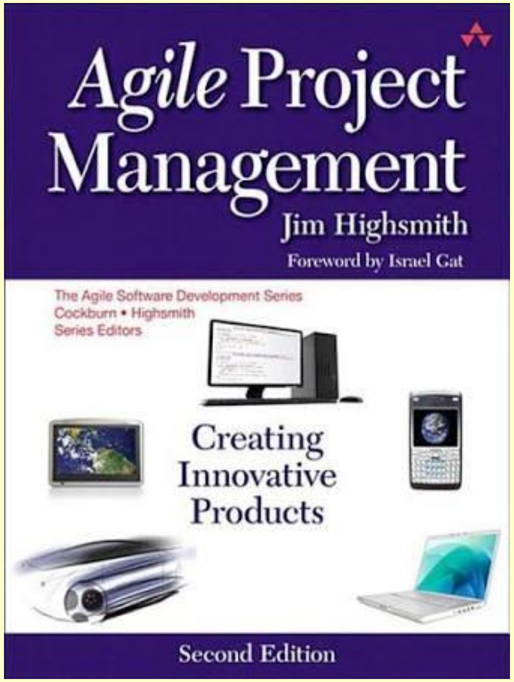
Agile Project Management

Jim Highsmith

Chapter 5

An Agile Project Management Model

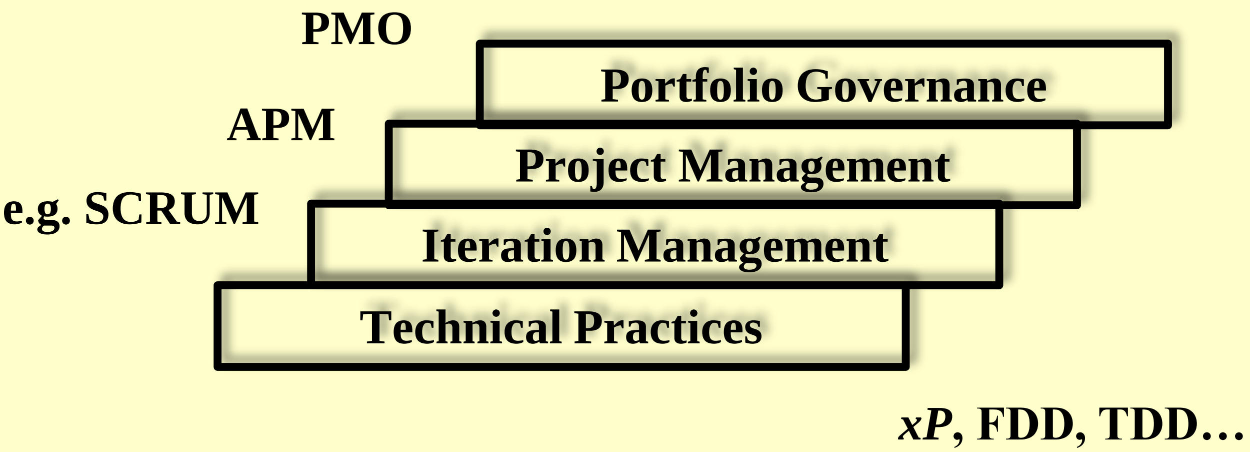


"We improve effectiveness and reliability through situationally specific strategies, processes, and practices."
One of the 6 Principles of the Declaration of Interdependence

Another way of saying "one size does *not* fit all."

Differences do exist between large organizations as well as multinational organizations.

Agile Enterprise Framework (Fig. 5-1)



PMO

Portfolio Governance

APM

Project Management

e.g. SCRUM

Iteration Management

Technical Practices

xP, FDD, TDD…

Portfolio Governance (Chapter 12)

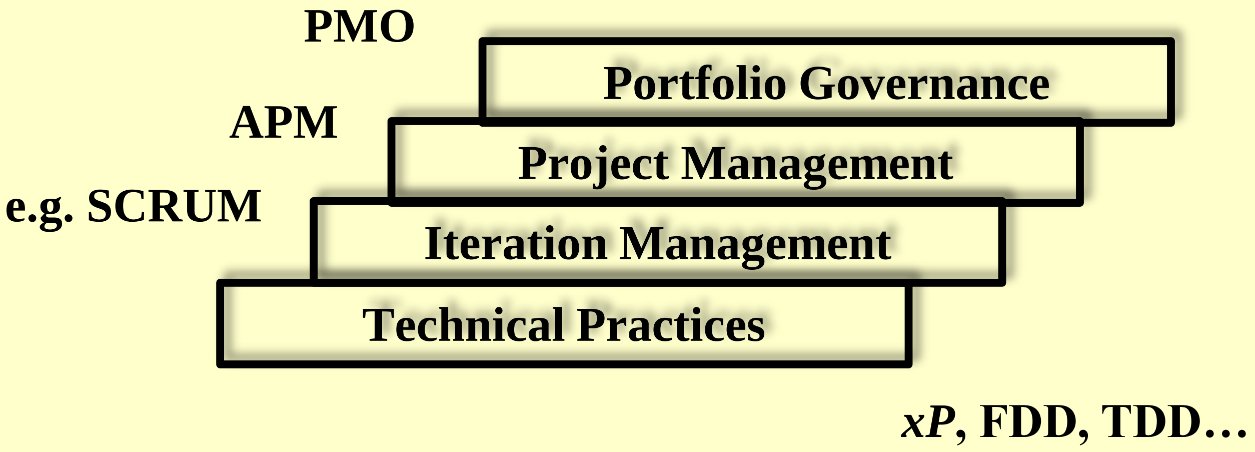
Common framework for executive evaluation of projects

… addressing investment (ROI), risk (uncertainty), progress

Project Management

Focus on overall project/release activities, assisting coordination among multiple feature teams, and managing the project externals (… sample source: PMBOK)

Agile Enterprise Framework (Fig. 5-1)



Iteration Management

Focus on planning, execution, and team leadership during short individual iterations.

Technical Practices Layer

From continuous integration to pair programming, from Test Driven Development to refactoring.

“Continuous integration and ruthless automated testing are core practices that should not be left out.

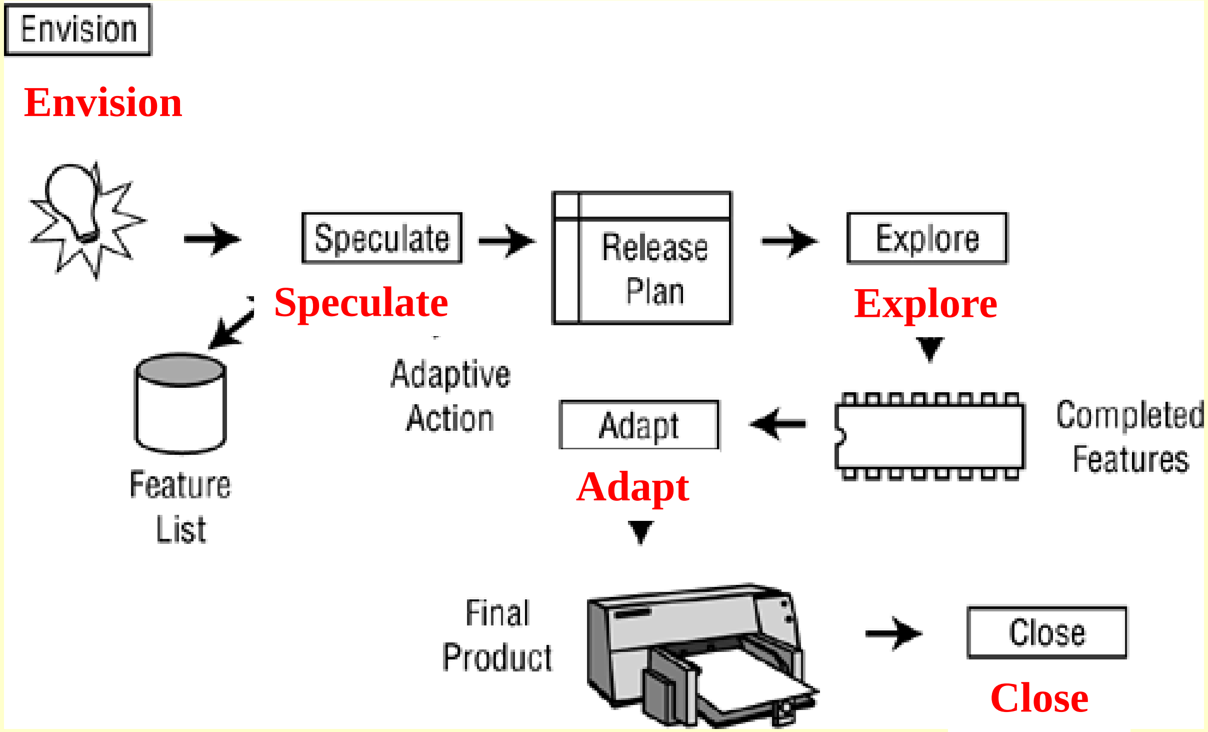
Agile Delivery Framework

Reliable innovation requires a process framework that is organic, flexible and easy to adapt.

The framework **must**:

- Support an envision, explore, adapt culture
- Support a self-organizing, self-disciplined team
- Promote reliability and consistency to the extent possible given the level of project uncertainty
- Be flexible and easy to adapt
- Support visibility into the process
- Incorporate learning
- Incorporate practices that support each phase
- Provide management checkpoints for review

APM Delivery Framework (Fig. 5-2)



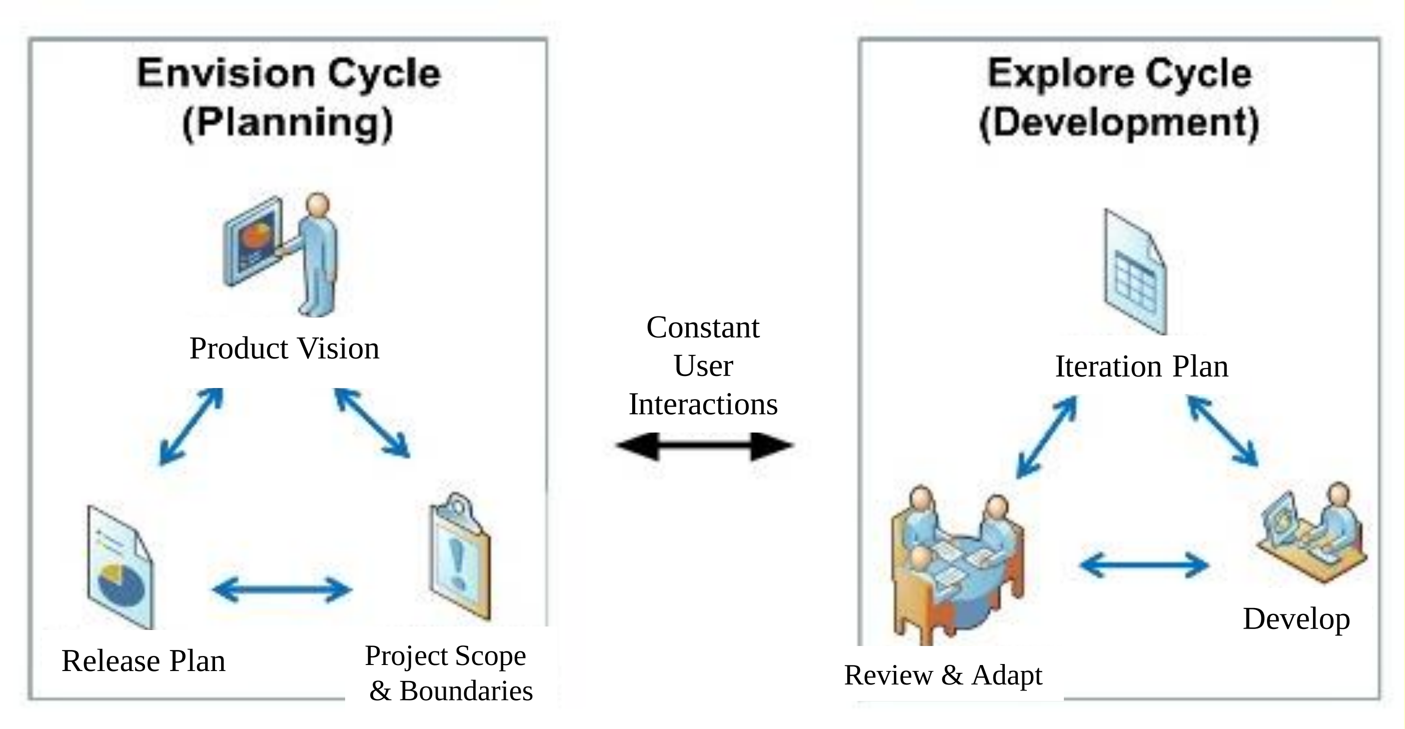
Envision: Vision is critical at the outset

Speculate: Not Planning which represents prediction & relative certainty

APM model replaces the Design, Build, Test phases with **Explore** (meaning nonlinear, concurrent, non-waterfall)

Close objectives being knowledge transfer and celebration

APM'S **Envision** & **Explore** Cycles



Emphasis is on Cycles… not unidirectional flow

All parts of the Envision Cycle may be executed multiple times during a project (iterations)

Critical Success Factors

Early in a project… **Envision Phase**

Team **envisions** what to deliver… vision and scope

In addition envisioning who will be involved

- the community of customers
- product managers
- team members
- stakeholders

Team **must** **envision** how they intend to work together!

Speculate Phase

“to conjecture something based on incomplete facts or information”

“People believer when they plan that they introduce certainty which is far from the truth.

… what they introduce is something to gauge their performance by

… when the gauge does not reflect reality, they fail to replan.”

Speculate Phase

Consists of:

- Gathering the initial brad requirements for the product
- Defining the workload as a backlog of product features
- Creating a iterative, feature-based release plan
- Incorporating risk mitigation strategies into the plan
- Estimating project costs and generating other required administrative and financial information.

Explore Phase

This phase delivers product stories

Three **critical** activities are involved:

1. Delivering planned stories by managing the workload and using appropriate technical practices and risk mitigation strategies.
2. Creating a collaborative, self-organizing project community, which is everyone’s responsibility but is facilitated by the project and iteration leaders.
3. Managing the iterations among customers, product management, and other stakeholders

Adapt Phase

“Adapt” implies modifications or change rather than success or failure.

> “Responding to change is more important than following a plan”
>
> *Manifesto*

Results are reviewed from customer, technical, people and process performance and project status perspectives!

Considering actual versus a revised outlook on the project given up-to-minute information.

After the **Envision** Phase, the loop is Speculate-Explore-Adapt, with each iteration successively refining the product

Close Phase

Key objective of the “Close” Phase and the “mini” close at the end of each iteration:

Learning and incorporating that learning into the work of the next iteration … or passing it on to the next project team.

NOTE

Not included in this framework description… two important phases:

1. the early conceptualization phase
2. the later deployment phase.

Selecting and Integrating Practices

- Practices are just practices… various ways of carrying out some goal.
- Practices are only good within some context, which might include
  - Principles
  - Problem Type (E.G. Exploratory)
  - Team Dynamics
  - Organizational structure.

“There is no such thing as a best practice. There are only good practices for a given situation.”

Principles guide Practices, Practices instantiate Principles”

Guiding principles:

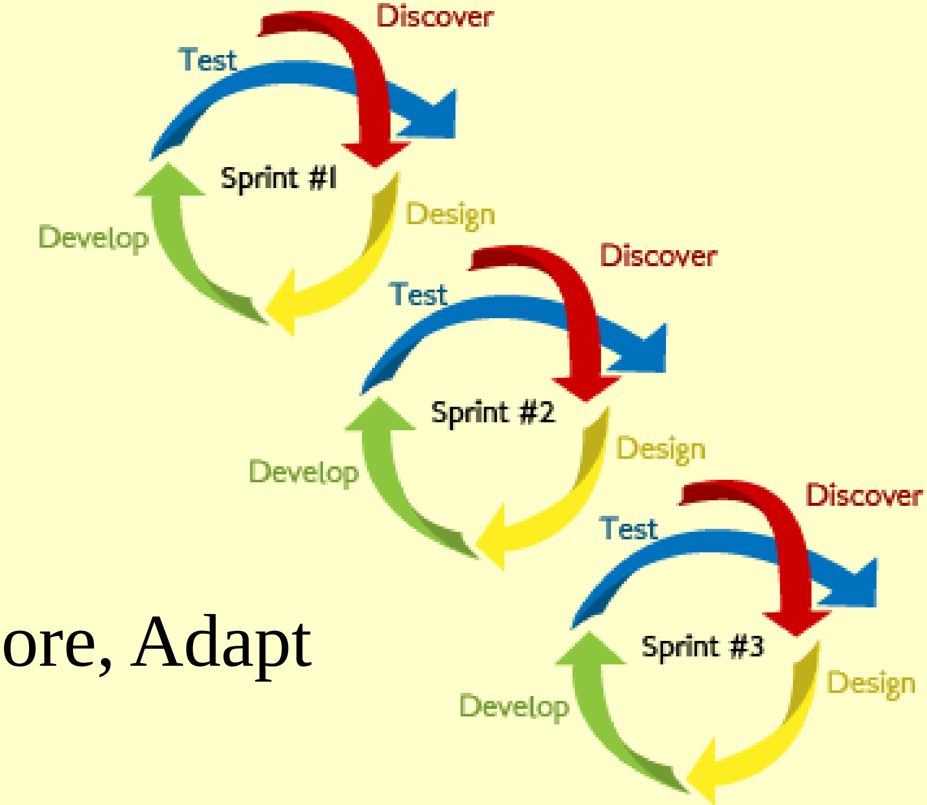
- Simple
- Generative [1], not prescriptive
- Aligned with agile values
- Focused on delivery (value adding), not compliance
- Minimum (just enough to get the job done)
- Mutually supportive (a system of practices)

[1] Generative practices are the minimal set that works together as a system… not prescribing everything but representing practices that are of high value.

“Judgment Required”

“Waterfall” style models imply linearity and repeatability (initiate →Plan→Define→Build→Test)

APM terms were selected to imply iterative evolution.



Discover ≡ Speculate, Explore, Adapt

Common questions

What about planning, architecture and requirement’s phases?

- These are activities, NOT phases
- These activates are included … but spread across iterations

What of the risk of rework if initial architecture work overlooks what turns out to be critical?

- The cost is less in that changes are made as needed throughout the APM cycle.

All work should be evolutionary, even architecture development. Getting up-front architecture wrong in serial development usually means poor long-term adaptability because no one can stomach changing architecture late in a project

Project Size???

“A 500-person team can’t be as agile as a 10-person team, but…

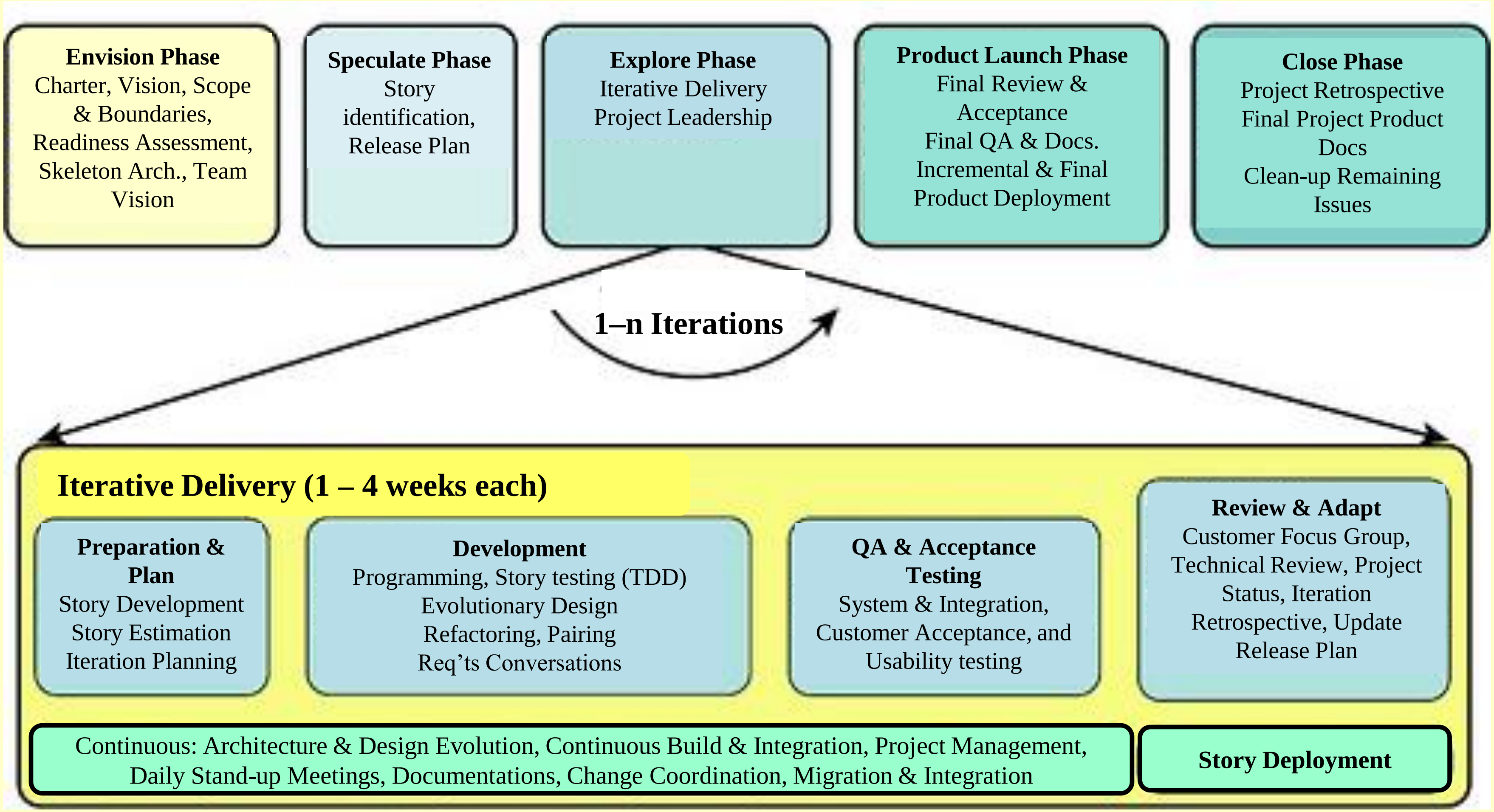
… it can be more agile than a competitor’s 500 person team.

By focusing on value, delivery, self-organization, and self-discipline… even larger teams burdened with complex coordination issues can readily adapt to business, technology, and organizational changes”

What’s the probability of this happening?

Is there any evidence to support this “claim”?

Expanded APM Delivery Framework (Fig. 5-4)



Envision Phase
Charter, Vision, Scope & Boundaries, Readiness Assessment, Skeleton Arch., Team Vision

Speculate Phase
Story identification, Release Plan

Explore Phase
Iterative Delivery
Project Leadership

Product Launch Phase
Final Review & Acceptance
Final QA & Docs.
Incremental & Final Product Deployment

Close Phase
Project Retrospective
Final Project Product Docs
Clean-up Remaining Issues

1–n Iterations

Iterative Delivery (1 – 4 weeks each)

Preparation & Plan
Story Development
Story Estimation
Iteration Planning

Development
Programming, Story testing (TDD)
Evolutionary Design
Refactoring, Pairing
Req'ts Conversations

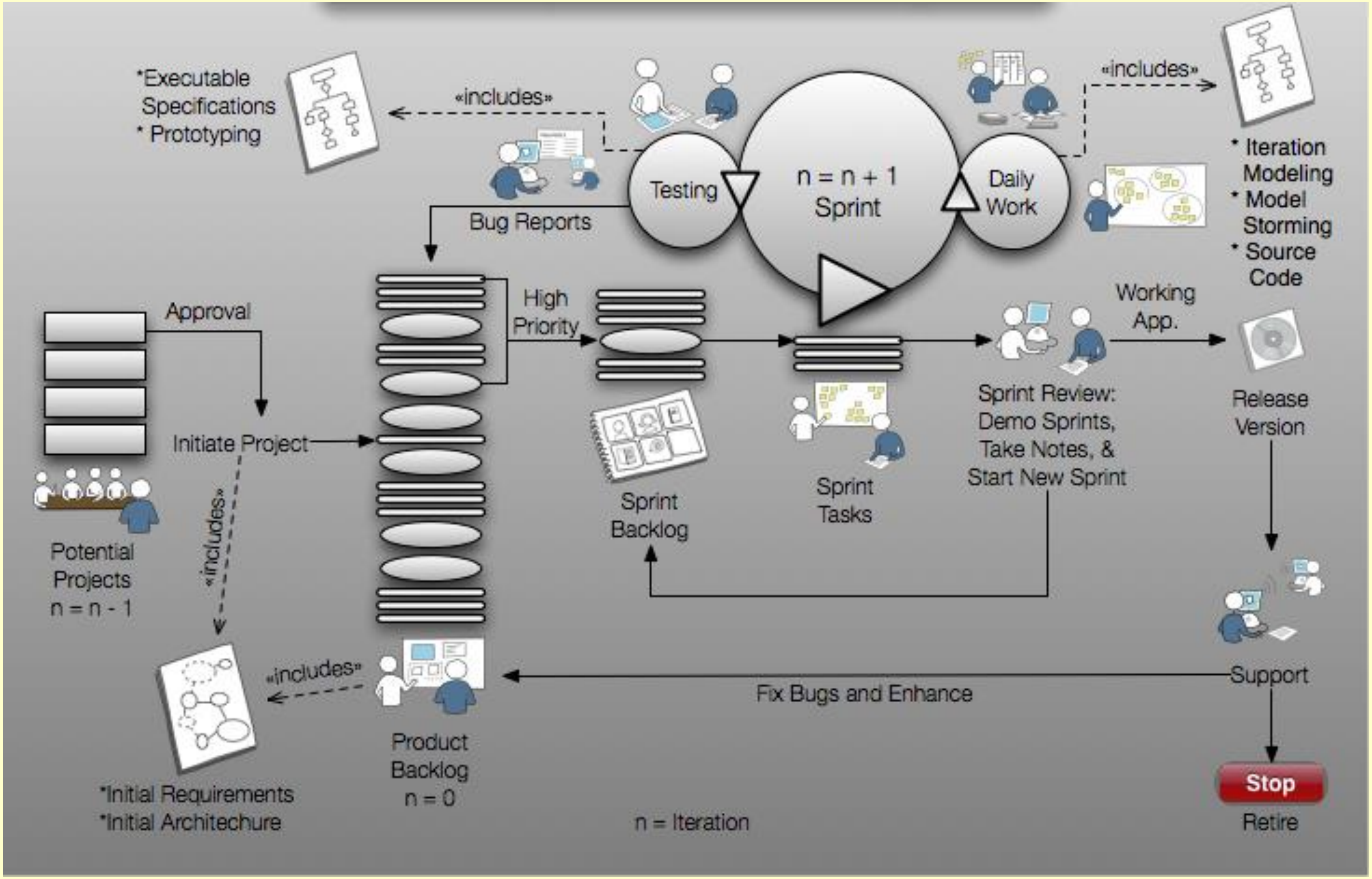
QA & Acceptance Testing
System & Integration, Customer Acceptance, and Usability testing

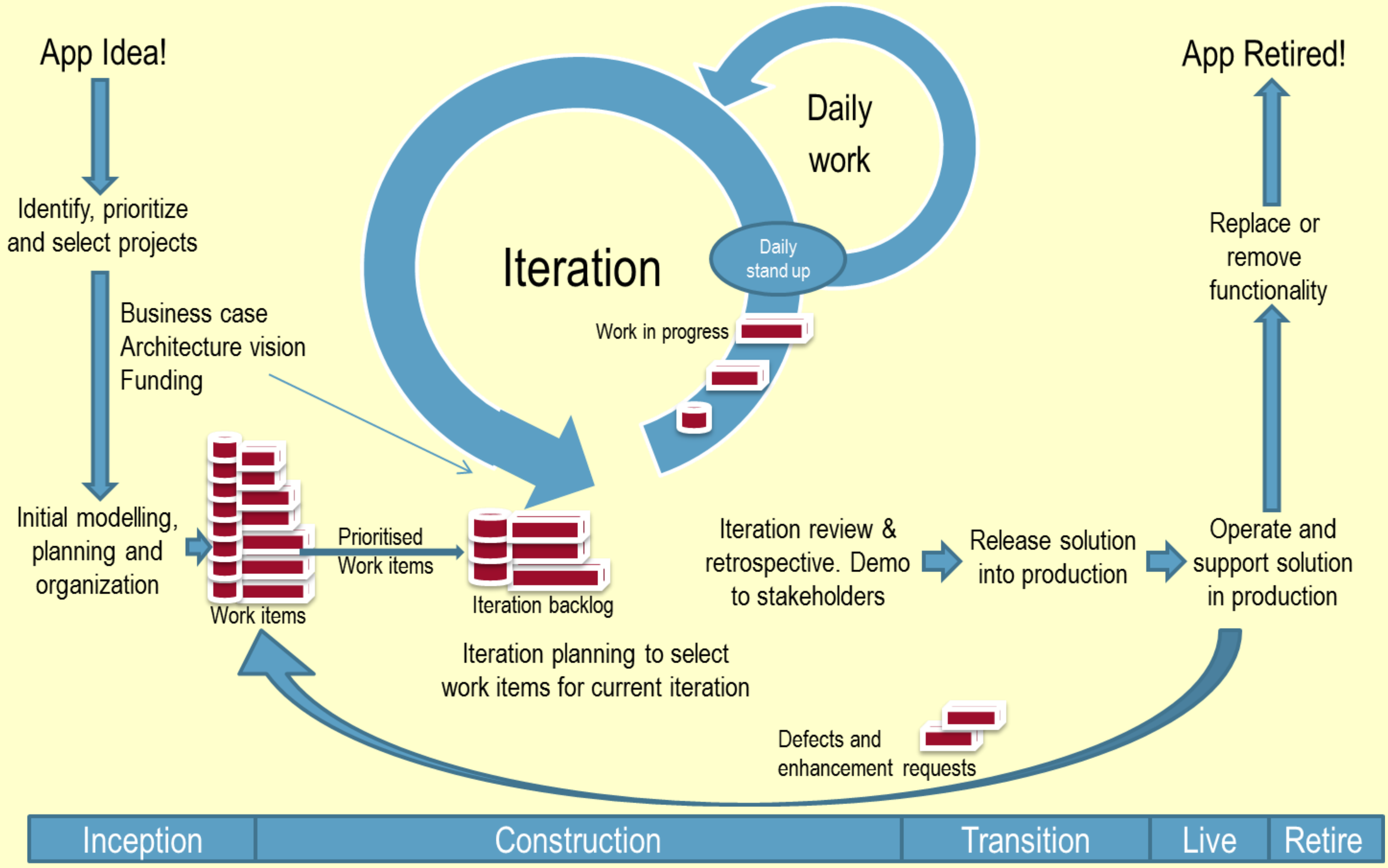
Review & Adapt
Customer Focus Group, Technical Review, Project Status, Iteration Retrospective, Update Release Plan

Continuous: Architecture & Design Evolution, Continuous Build & Integration, Project Management, Daily Stand-up Meetings, Documentations, Change Coordination, Migration & Integration

Story Deployment

Agile & SCRUM Development Lifecycle





App Idea!

Identify, prioritize and select projects

Business case
Architecture vision
Funding

Initial modelling, planning and organization

Work items

Prioritised Work items

Iteration backlog

Iteration

Daily work

Daily stand up

Work in progress

Iteration planning to select work items for current iteration

Iteration review & retrospective. Demo to stakeholders

Release solution into production

Operate and support solution in production

Replace or remove functionality

App Retired!

Defects and enhancement requests

Inception | Construction | Transition | Live | Retire