An Agile Accounting Model: Key to Enterprise Agile

Pat Reed & Walt Wyckoff

August 15, 2012

Overview

What ‘s this session about?

- Perspectives: de-mystifying Agile project cost accounting
- Patterns: mapping current accounting standards to an Agile project accounting model
- Partnerships: building relationships and co-creating a practical, scalable and defensible solution

How will do we that?

- Lightning talks and discussions (20 minutes)
- Brainstorming “breakout” workgroups (20 minutes)
- Shared learnings and conversations (15 minutes)
- Guided group solutioning, review and Q&A (35 minutes)

Key Learnings:

- A deeper understanding of accounting principles and perspectives
- Practical ideas for solving Enterprise level agile accounting and related problems

Background

- Explosive and strategic growth of technology in the 90's
- Recognition that internal software developed via Agile should be capitalized differently
- Mandatory accounting guidelines require:
  - Organizations to define and report IT development costs
  - Standardize financial reporting
  - Reduce inconsistencies across companies
  - ***Better enable investors to make informed decisions***
- Guidelines were designed around a phased, waterfall based IT investment model
  - IT projects delivered in discrete preliminary, development and post implementation phases
- Agile delivery models have inherent friction with legacy accounting practices
- Before Agile can scale:
  - The accounting issue needs more clarification
  - Agile software and project accounting practice needs to be developed

Accounting Background

1. SEC requires that GAAP be followed
2. FASB is the highest authority in establishing generally accepted accounting principles for public and private companies, as well as non-profit entities
3. AICPA SOP 98-1 and GAAP ASC 350-40 Internal Use Software provide guidance on how to apply GAAP principles and are written in language designed around a waterfall SDLC
4. Corporate accounting policies have been applied using the SOP 98-1 language
5. Accountants value accuracy and predictability and embrace principles of conservatism and consistency
6. Accountants don't understand Agile

Software Capitalization – What is it?

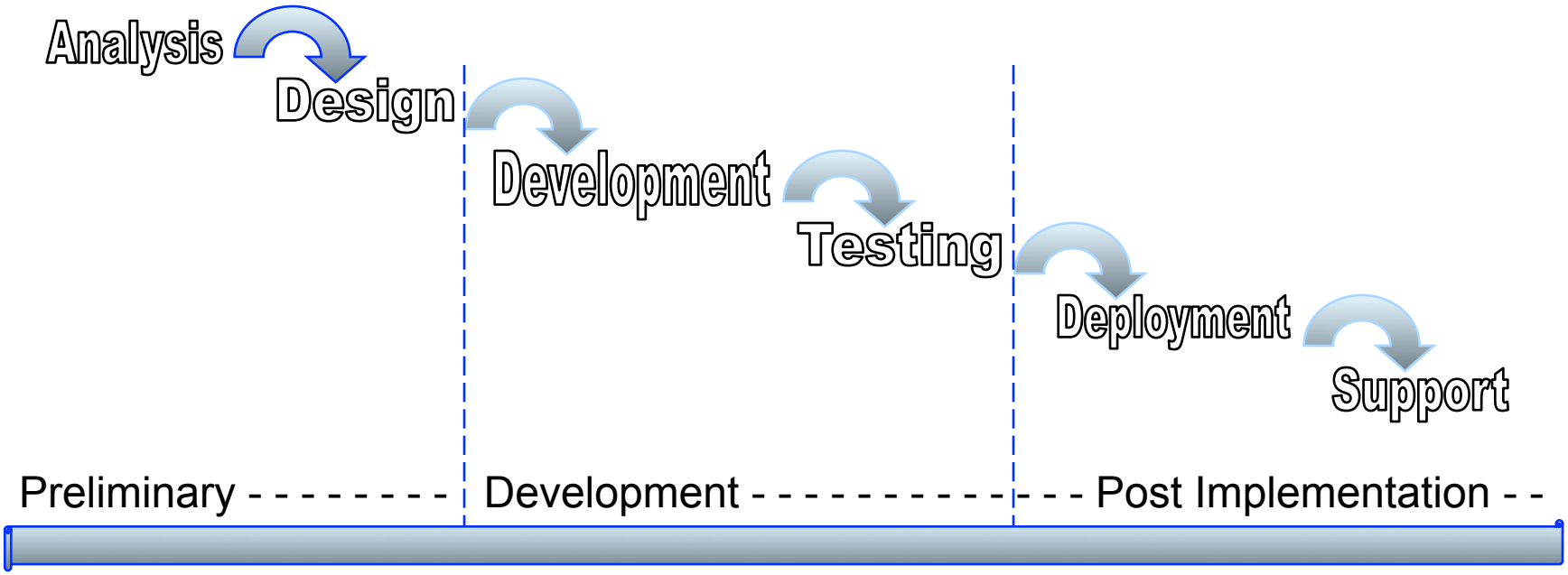
1. Software developed internally can be classified as an asset
   a. Can be capitalized if it will be used in production for more than one year
   b. The cost of a capitalized asset is depreciated over a number of years
   c. Software that has been expensed is not classified as an asset
2. Depreciate the value of the software asset over time
   a. Corporations are taxed on profits – depreciation costs reduce tax year profits
   b. Reduces tax liability
3. What can get Capitalized
   a. Software in beneficial use substantially beyond the end of the current tax year
   b. New software that has a useful life substantially beyond one year
   c. Improvements that prolong the life of the software
   d. Adaptations that permit the software to be used for a new or different purpose

Relevant GAAP

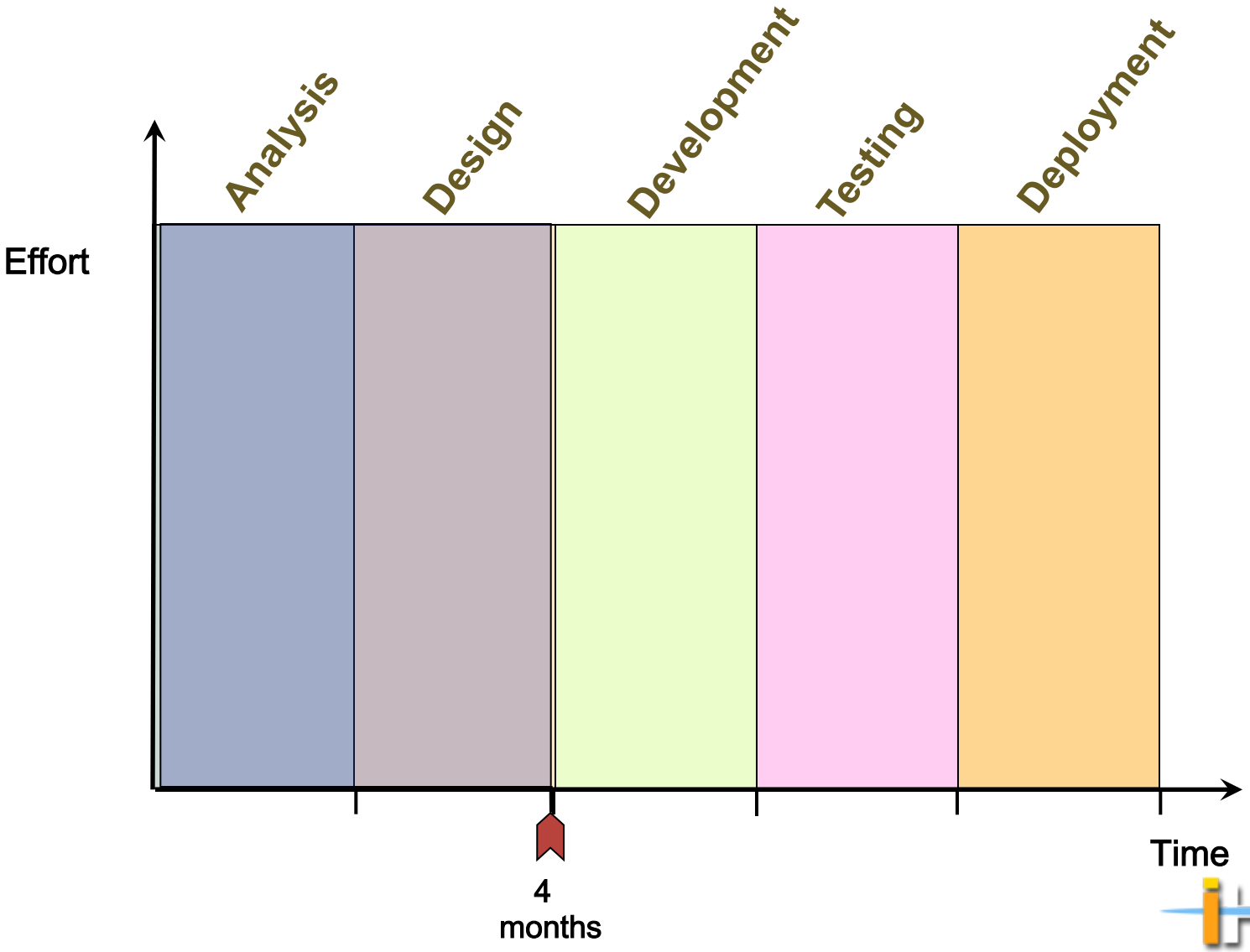
1. Objectivity principle: the company financial statements should be based on objective evidence.
2. Materiality principle: the significance of an item should be considered when it is reported.
3. Consistency principle: The company uses the same accounting principles and methods from year to year (note: referenced by documented policies)
4. Conservatism principle: when choosing between two solutions, the one that will be least likely to overstate assets and income should be picked

What's The Challenge?

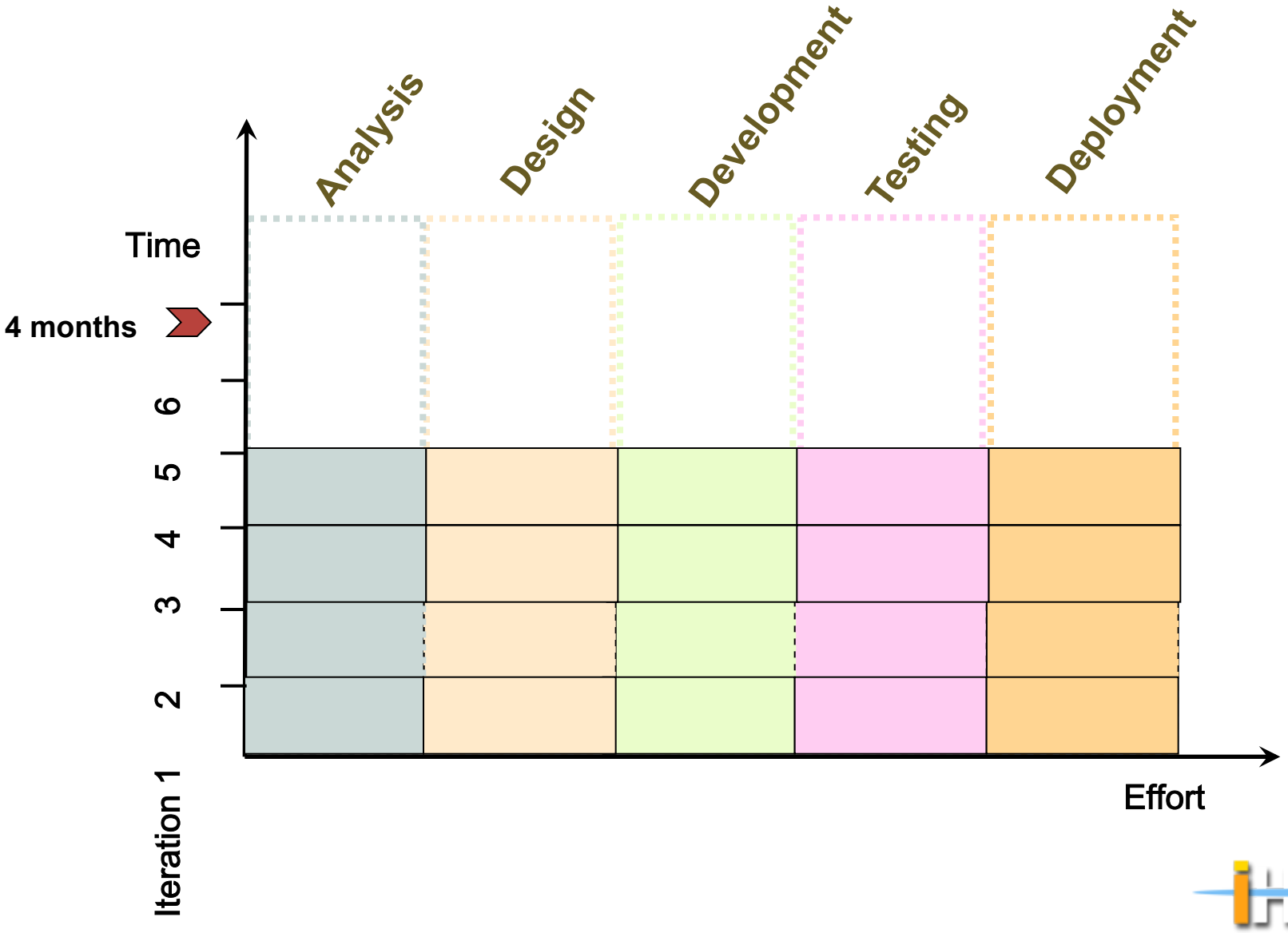
1. To ensure compliance, IT managers must estimate, allocate, track and report Agile project labor costs to internal IT projects based on project work done in three specific phases: Preliminary, Development and Post Implementation
2. Organizations utilizing a traditional waterfall model can readily adapt their labor and project costing to the guidelines using the following framework:



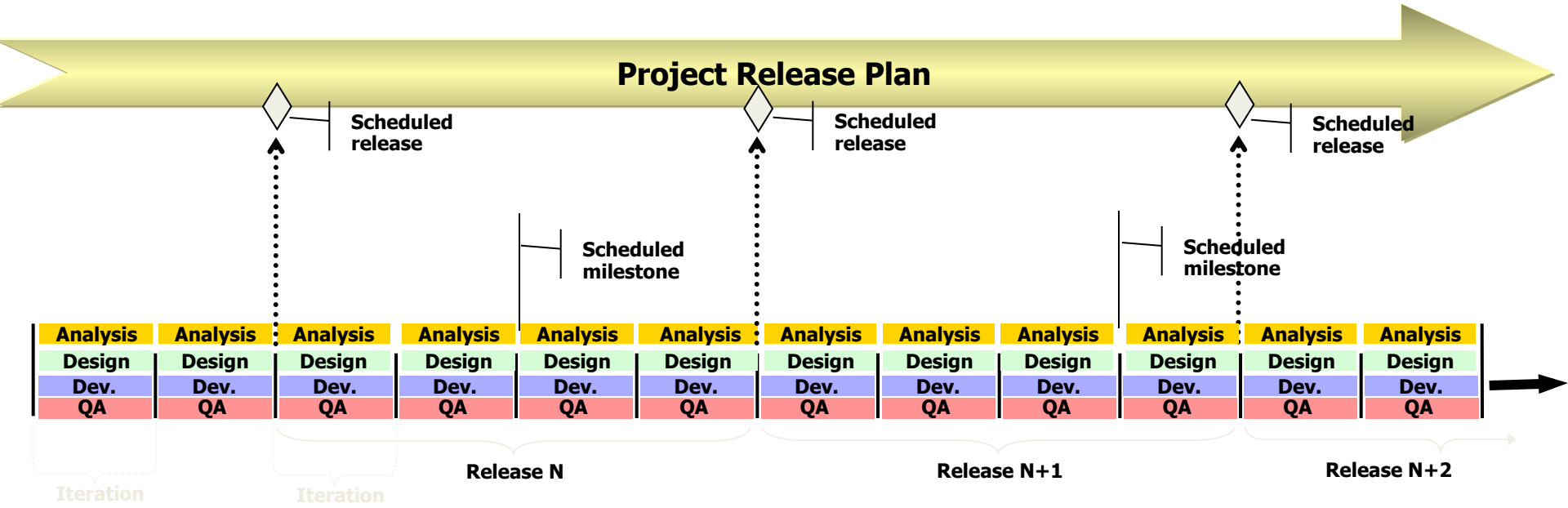
Waterfall Methodology



Agile Methodology



Agile Development Roadmap



Q - How can Agile organizations ensure compliance and accurately estimate, allocate and track labor costs to IT projects based on project work done in three specific phases when analysis, design, development and QA work is continuously repeated and intertwined throughout each iteration?

A – Partner with IT Finance / Technical Accounting / Financial Reporting / Audit to customize a simple, defensible and sustainable solution that's consistent with Agile Principles, Accounting Principles, SOP 98-1 and ASC 350-40

“Problems cannot be solved by the same level of thinking that created them.”

Albert Einstein

“What we know sometimes gets in the say of what we need to know.”

Pat Reed

Exploratory Ideas

1. Puzzles, Patterns & Paradigms
   - As Agile "experts" – why do we fall into the same "trap"?
   - Idea: reframe our agile mental models
2. Principles & Partnerships
   - Why is Agile Accounting a point of friction?
   - How can we help accountants understand Agile?
   - Idea: Advocacy – let's understand accounting principles & CSF's
   - Idea: Partnership - Creating a shared language and culture – and co-create a solution
3. Clarify the benefits of Agile
   - Idea: In an accounting and financial ROI context
   - Idea: Map GAAP to Agile practices

“The seeds of change are in the questions we ask”

Pat Reed

Brainstorm

Activity:

From an Accountant's or CFO's perspective…

1. Walk through SOP 98-1 and draft an Agile Accounting Solution.
2. Talk through the GAAP principles of conservatism and consistency…and discuss the importance of accuracy, predictability and risk mitigation. Compile a series of questions that reflect what an accountant and CFO need to know about Agile.
3. Design talking points to present to your CFO to "sell" Agile. How can you help them understand the benefits of Agile re: GAAP?

What's Important to Your Accountant, and What They Need to Know About Agile

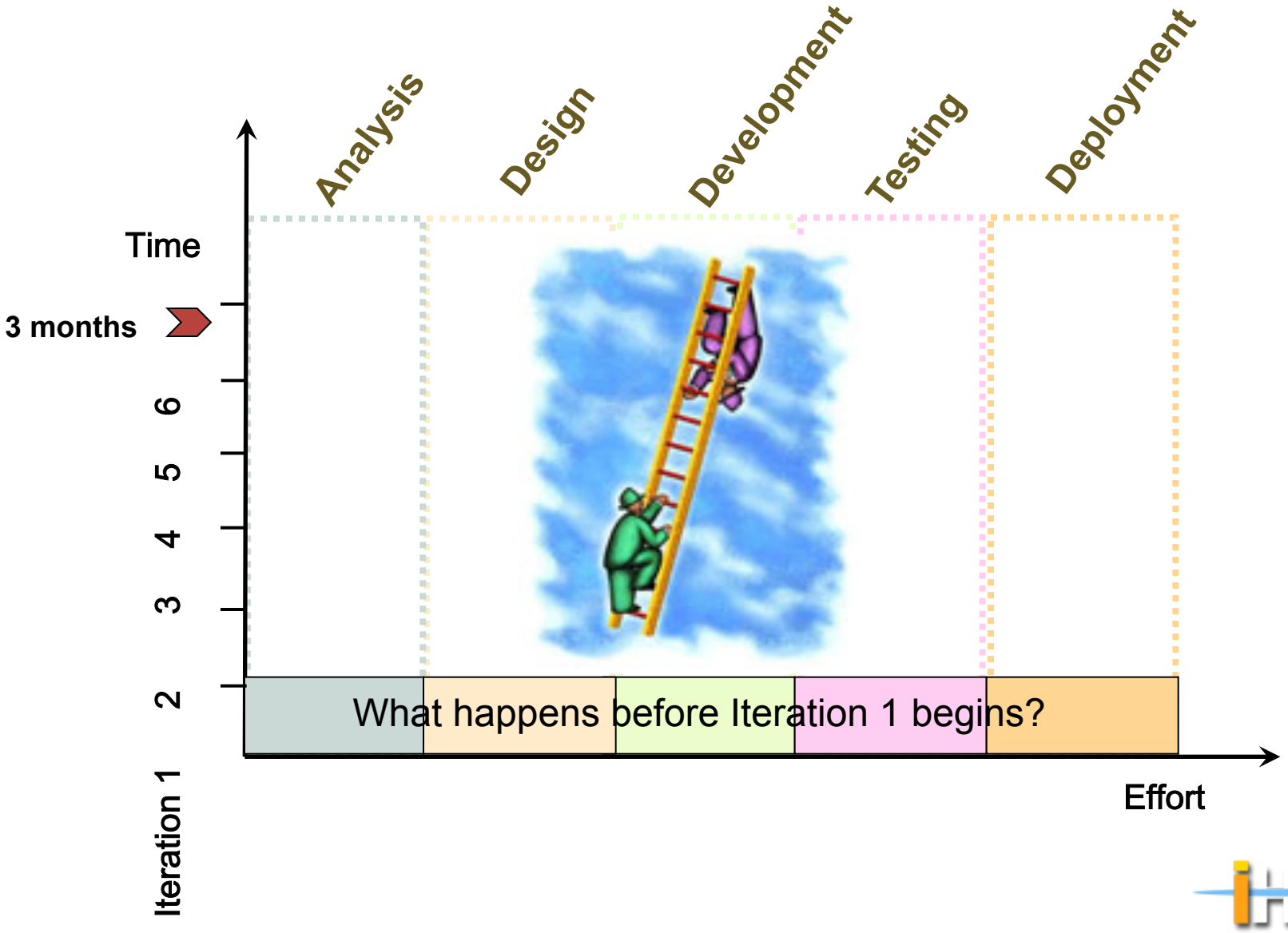
Pitching Agile to Your CFO

Deeper Dive into Agile Accounting

Critical Success Factors or Tests

- Simple
- Defensible / Auditable
- Consistent with SOP 98-1 and GAAP
- Scalable
- Sustainable
- Easy to understand, interpret, implement, administer
- Value and risk balanced

Agile Accounting



Agile Project Stages

- The Preliminary Project: " **What** "
- The Development Stage: " **How** "
  - design of software, configuration and interfaces
  - coding, retrospectives, testing
  - Hardware installation and preparing software for intended use
- **"When" Does The Development Stage Start & End**
  - Design Storming is the clear bright line between the preliminary (what) and the start of development (how) phases. Auditable deliverable: capture and store e-mail as evidence of management approval to proceed to development
  - 72 hours after production build and final user acceptance testing is the clear bright line for development to end and the "post-implementation" phase to begin (note: during these 72 hours, the development teams are on point along with the Release team and only after the 72 hours, transfers to production support).

Key SOP 98-1 Guidelines

- 3 Stages of an IT project:
  - Preliminary Stage – Costs must be expensed
  - Application Development Stage – Most costs should be capitalized
  - Post Implementation Stage – Costs must be expensed
- Capitalization begins when
  - the preliminary project stage is completed and
  - management, with the relevant authority authorizes and commits to funding a computer software project
  - it is probable that the project will be completed and the software will be used to perform the function intended.
- Capitalization ends no later than the point at which a computer software project is substantially complete and ready for its intended use.

An Agile Approach . . .

Step 1: Paradigm shift: let's suspend our knowledge of the differences between Agile and Waterfall.....and take a new look at the "waterfall centric" project stage framework through an Agile lens

Step 2: Adopt an Agile Project Stage Framework

Step 3: Standardize on a lightweight process to capture evidence of management authorization and commitment to project funding

Step 4: Apply and adapt this framework to your organization

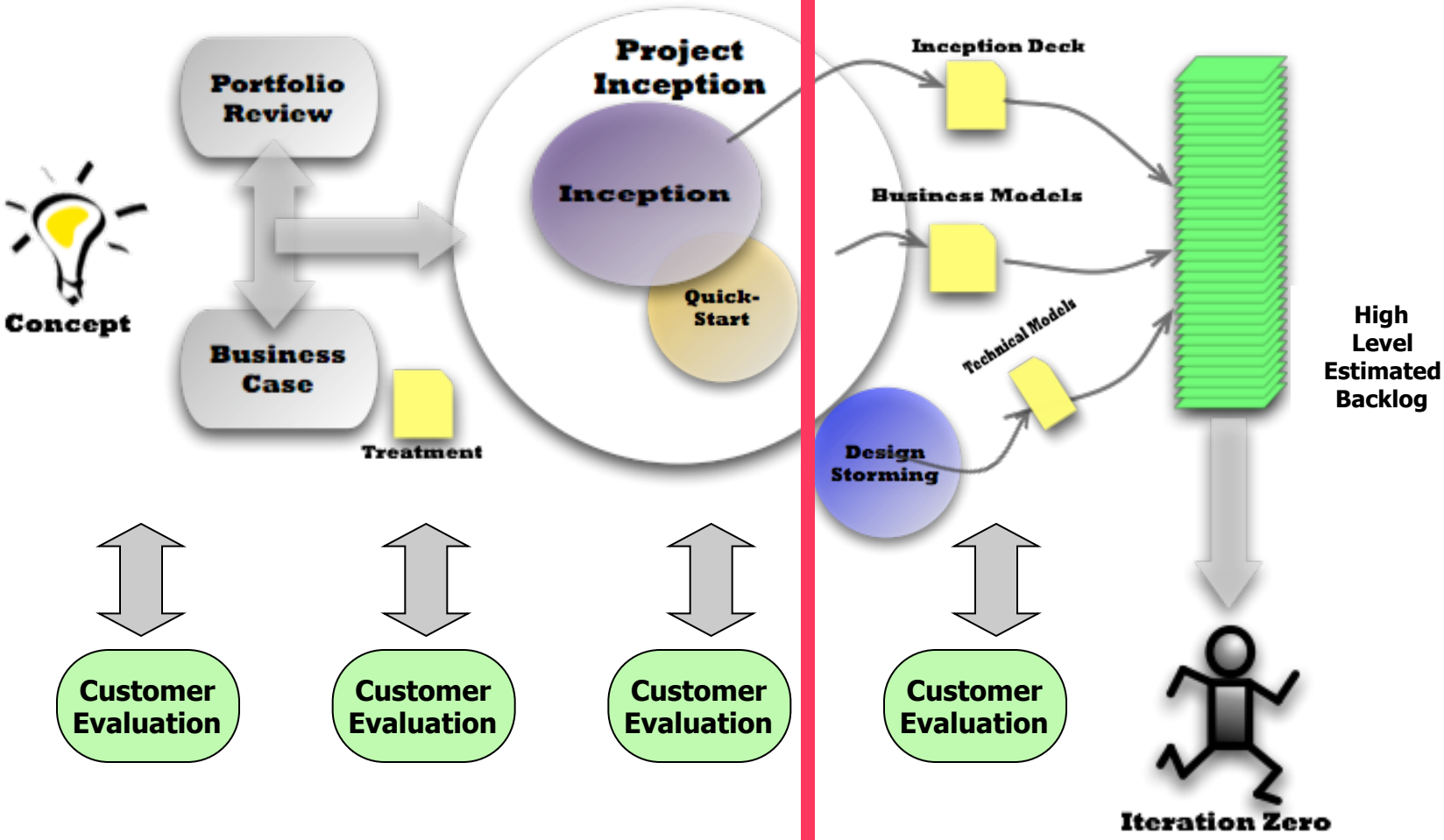
Scorecard

- Does this solution clearly address the 3 stages of an IT project?
  1. Preliminary (expense)
  2. Development (mostly capital except for administration, overhead, training and data conversion costs)
  3. Post Implementation (mostly expense)
- Have we documented management authorization of funding?
- Have we assessed probability that the project will be completed and resulting software used to perform the function intended?
- When do we define that the software is complete and ready for it's intended use?
- Would this proposed solution provide a defensible, auditable, scalable and sustainable solution?
- Is it consistent with GAAP and FASB (SOP 98-1)?

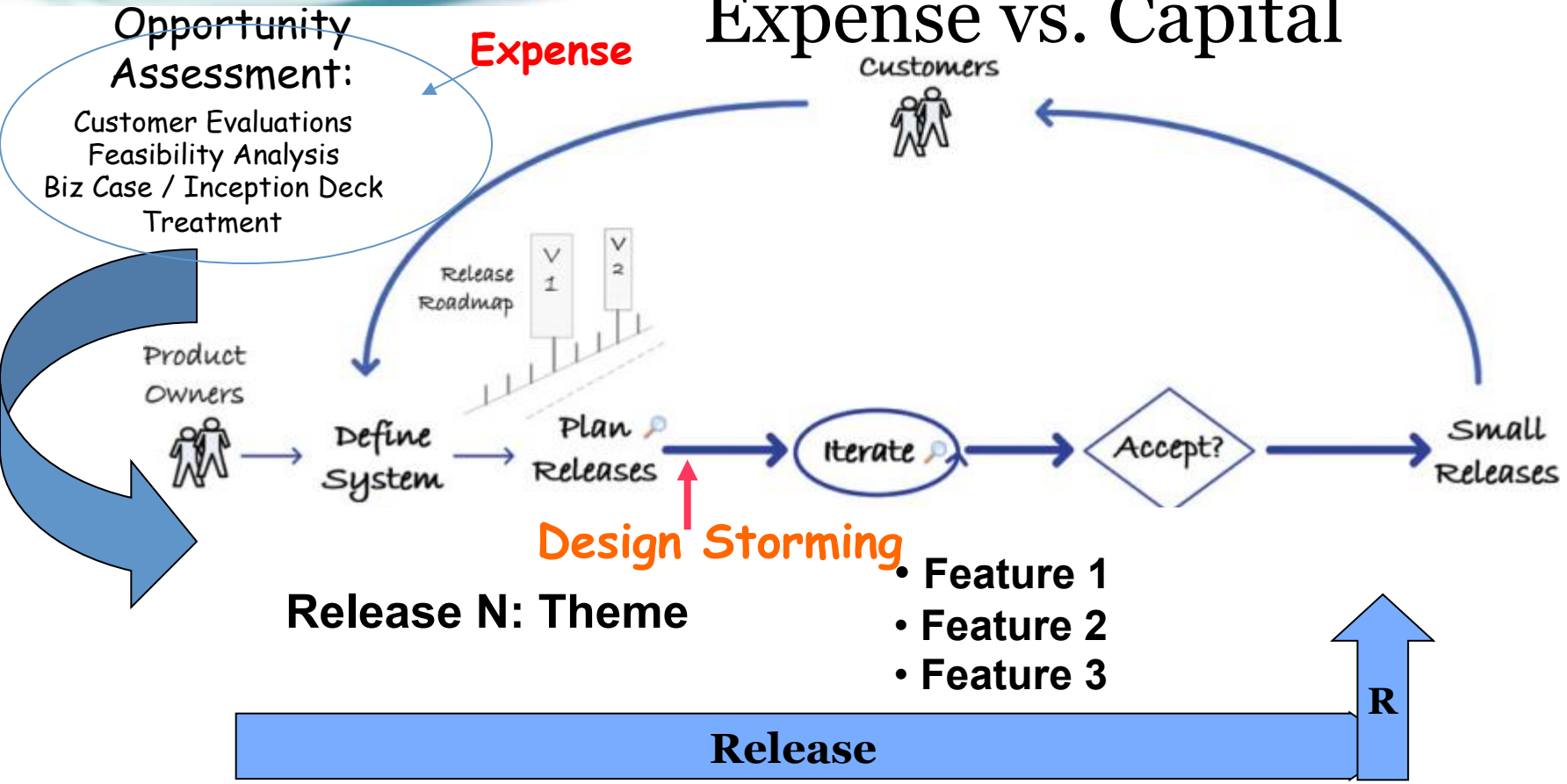
Agile Project Stages

- The Preliminary Project Stage is analogous to research and development activities, primarily expense and covers the conceptual formulation and evaluation of alternatives, determination of existence of needed technology and final selection of alternatives (The What).
- The Development Stage covers all work related to The How: design of software, configuration and interfaces, coding, retrospectives, installation to hardware, testing (including parallel processing) and all work preparing the software for its intended use.
- Standardize on a clear bright line between the preliminary (what) and the development (how) phases, and capture evidence of management approval (which marks the start of capitalizable work): Design Storming
- Standardize on an equally clear bright line that marks the end of development which concludes when all iterations of the project are complete and the asset is ready for its intended use and placed in service (which includes all deployment work necessary to get the software into production as well as production testing, acceptance and stabilization): 72 hours after production implementation and final user acceptance testing.

Stage 1: Opportunity Assessment



Expense vs. Capital



Release N: Theme

- **Feature 1**
- **Feature 2**
- **Feature 3**

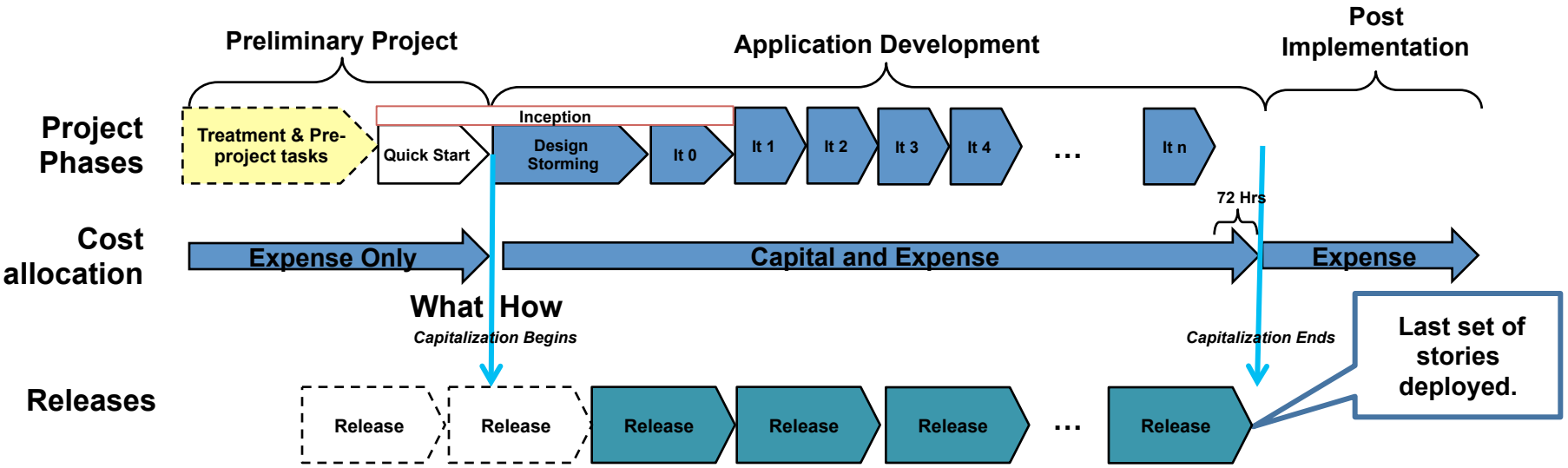
Release

R

Backlog	Iteration 1	Iteration 2	Iteration 3	Iteration …	Backlog
• Story 1 • Story 2 • Story 3 • Story …	• Story 1 • Story 2	• Story 3 • Story 4	• Story 5 • Story 6 • Story 7	• Story 8 • Story 9 • Story 10	• Story 11 • Story 12 • Story …

Capital

Costs can be Capitalized once the “Approval to Start” has been approved and end at the completion of the Application Development stage.



- The Preliminary Project Phase: “**What**“ **(Inception & Quickstart)**
  - When the project team has completed the feasibility analysis and the review of the “epic” stories is complete, the Preliminary Project phase is complete.
- The Development Stage: “**How** “
  - design of software, configuration and interfaces
  - coding, retrospectives, testing
  - Hardware installation and preparing software for intended use
- **“When” Does The Development Stage Start & End**
  - Design Storming is the clear bright line between the preliminary (what) and the start of development (how) phases. It’s important that TM’s capture and store e-mail as evidence of management approval to proceed to development
  - 72 hours after the last production implementation, final user acceptance testing and Site Ops handoff is the clear bright line for development to end and post-implementation to begin.

Preliminary Project Costs…

- Strategic decisions to allocate resources between alternative projects
- Determining performance requirements (what do we need the software to do) and high level systems requirements
- Vendor product demonstrations
- Exploring alternatives (Buy vs. Build)
- Feasibility Analysis
- Vendor selection

Clear Bright Line

- Critical Success Factor: It's important to capture evidence of management authorization (with time and date stamps) to document transition from expense (preliminary phase) to capital (development phase) of the project
- This clearly marks the transition from the "What" to the "How"
- Memorializing and documenting this authorization is critical for future auditability

Stage 2: Development

- Most costs should be capitalized:
  - Designing the chosen path, including software configuration and software interfaces
  - Refinement of requirements (i.e. creation of detailed Use Cases or Stories, which have a lifecycle and evolve as they transform and mature)
  - Internal and external labor costs and fees associated with development (design, build, test, implement)
  - Costs to develop or obtain software that allows for access or conversion of old data by new systems
- Costs that Should be Expensed Include:
  - Training and Data Conversion (except as noted above)

Stage 3: Post Implementation

- Another clear bright line: 72 hours after implementation and production acceptance
- Post Implementation (Expense) examples include:
  - Support turnover and training
  - Certifying operational system
  - Post implementation review
  - Collecting and analyze process data
  - Ongoing Maintenance
  - Transformation / process reengineering / workforce restructuring

Discussion
Q&A

Useful References

- Link to SOP 98-1 document: http://efile.mpsc.cis.state.mi.us/efile/docs/14201/0031.pdf
- MindTools: http://www.mindtools.com/pages/article/newTMC_91.htm
- FASB - Financial Accounting Standards Board: http://www.fasb.org/st/
- Generally accepted accounting principles: https://en.wikipedia.org/wiki/Generally_Accepted_Accounting_Principles_%28United_States%29#Assumptions
- http://blog.lithespeed.com/2012/05/software-capitalization-and-agile.html
- http://stevedenning.typepad.com/steve_denning/2010/09/how-do-you-explain-agile-or-radical-management-to-a-cfo-.html
- http://itprojectfinancials.com/insights/2011/06/05/capitalizing-software-development-costs-from-sdcl-to-agile/

Retrospective

1. What are your observations and thoughts about today's session?
2. What new information was the most impactful for you
3. What puzzles you about the information presented today?
4. What obstacles do you see that could impact your use of the materials we covered today?
5. What ideas and recommendations do you have to share?

Agile Accounting

Pat Reed & Walt Wyckoff
preed@ihoriz.com
wwyckoff@ihoriz.com
www.ihoriz.com
800-542-8184

Appendix

SOP 98-1 Simple Rules

1. **The nature of work in the Development Phase determines whether it will be Capitalized or Expensed:**

Expense	vs.	Capitalization
What		How
People or Process-Centric		Asset-Centric
Administrative		Technical
Support		Decision-Authority
Discretionary/Supplemental		Asset-Critical

2. **Decision tree:**

IF
- Minimum expected life of 3 years beneficial use
- New software functionality
- Design/build/test cost results in the creation of a new asset of at least $100K cost

AND
- Completion of preliminary (expense) phase with e-mail from TM or PM to finance approval as evidence of readiness for design storming (triggering the development/capitalization phase)
- Any other Formal approvals of capital funding through design

AND
- High probability that the product will be completed as planned
- Work effort is directly related to asset /product design, development , testing or implementation/integration (except for administration, overhead, training and data conversion costs) (see WBS task detail for more clarification)

CAPITALIZE

ELSE
Expense

Accounting Glossary

- AcSEC – AICPA's Accounting Standards Executive Committee
- SEC – Security and Exchange Committee
- EITF – SEC's Emerging Issues Task Force
- GAAP – Generally Accepted Accounting Principles from the American Institute of Certified Public Accountants (AICPA)
- SOP 98-1 – American Institute of Certified Public Accountant's Statement of Position Accounting for the cost of Computer Software Developed or Obtained for Internal Use – issued on 2/27/1998 which all non governmental public or private organizations must follow and which are now part of GAAP. SOP 98-1 standardizes how organizations measure and report their investments

Agile Glossary

- **Quickstart –** High level Analysis and Requirements Capture
- **Customer Evaluations**
  - Identify customer preferences
  - Eliminate bad ideas
- **Inception Deck**
  - Charter: Organizational and Departmental Objectives
  - Business & Technical Outcome Vision (“What” & “How”)
  - In Scope / Out of Scope and Context Diagram
  - High Level Risks and Project Approach and Estimates
  - Stakeholder Agreements
  - Release Plan and Staffing Plan
  - Project Organization and Management Structure
  - Trade Off Sliders
- **Treatment**
  - Solution Description and Metrics of Success
  - High level functionality matrix and impact analysis
  - Assumptions and Questions
  - Estimated durations / costs / budget / contingency
  - High level resource plan