Agile Logic
Deliver Better Software Faster

Let Me Tell You What I Really Want
A User Story Writing Workshop

Agile / XP So. Cal. Meeting – 3/18/09

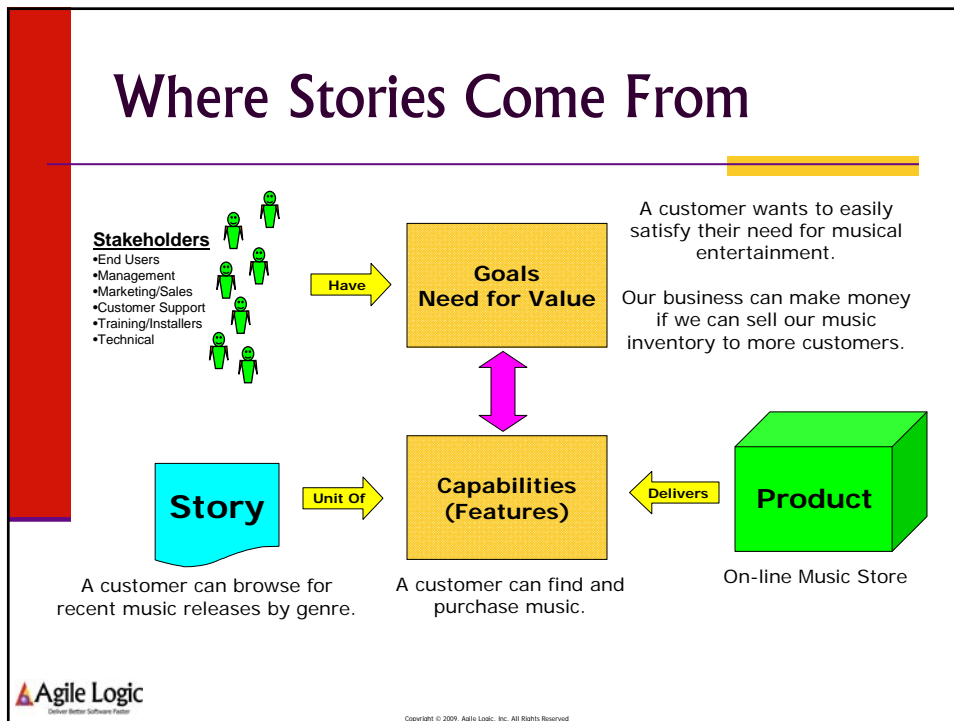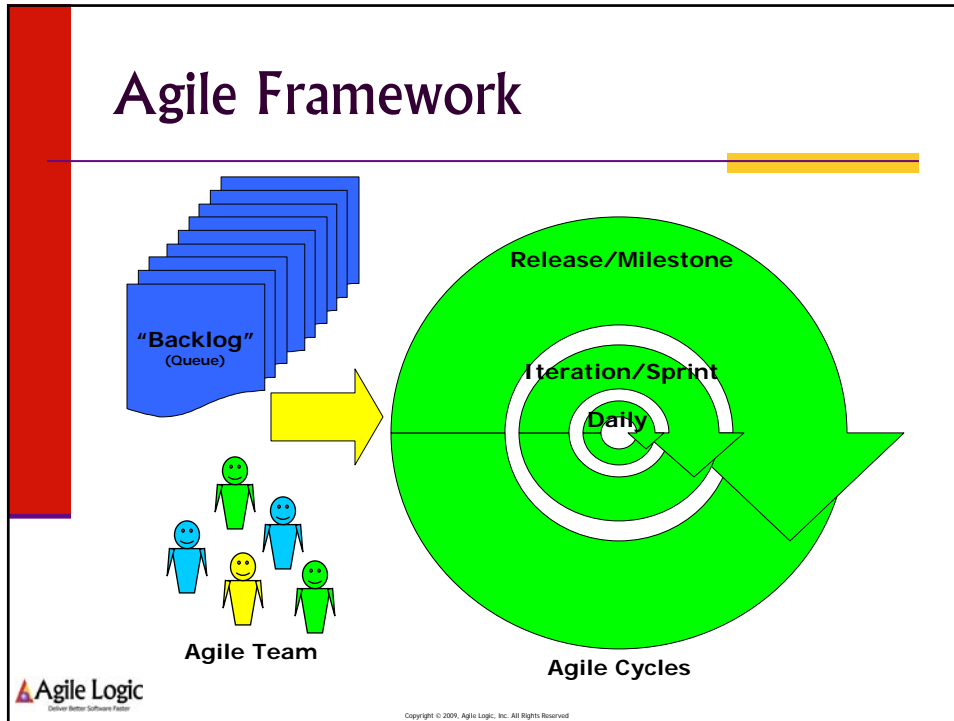Speaker: Paul Hodgetts, Agile Logic, www.AgileLogic.com

Rev 090330

Your Coach: Paul Hodgetts

- Team coach, trainer, consultant, developer
- Founder and CEO of Agile Logic (Fullerton)
- 26 years overall, 10 years agile experience
- Certified Scrum Trainer
- Worked with a lot of "enterprise" teams
- Author (Extreme Programming Perspectives)
- Speaker at conferences (Agile 200x, SD East/West, JavaOne)
- Active in Scrum Alliance, Agile Alliance (Program Director)
- Member of CSUF agile advisory board
- Contact info: phodgetts@agilelogic.com

Agile Logic
Deliver Better Software Faster

Agile Framework

"Backlog"
(Queue)

Release/Milestone

Iteration/Sprint

Daily

Agile Team

Agile Cycles

Copyright © 2009, Agile Logic, Inc. All Rights Reserved

Where Stories Come From

Stakeholders
•End Users
•Management
•Marketing/Sales
•Customer Support
•Training/Installers
•Technical

Have

**Goals
Need for Value**

A customer wants to easily satisfy their need for musical entertainment.

Our business can make money if we can sell our music inventory to more customers.

Story

Unit Of

**Capabilities
(Features)**

Delivers

Product

On-line Music Store

A customer can browse for recent music releases by genre.

A customer can find and purchase music.

Copyright © 2009, Agile Logic, Inc. All Rights Reserved

Basic Story Format

<Target Stakeholder> can <Capability> <Essentials>

A customer can browse for recent music releases by genre.

> A clear statement of the capability that must be present in the product for the stakeholder to gain value.

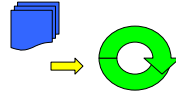> Stories are written collaboratively by the team.

Hands-On

Write Some Stories

OK, What Is A Story, Really?

A story is a "catalog entry."
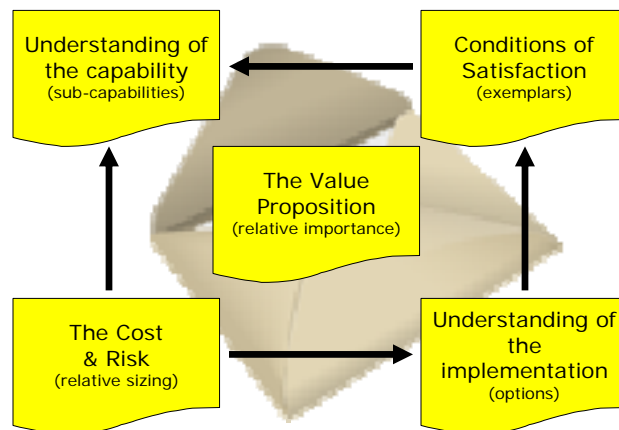
> A token for our conversation and learning about it

> A chunking mechanism for delivering in cycles

> An "envelope" for the details...

> The story statement is the title of the catalog entry
(What's written on the envelope)
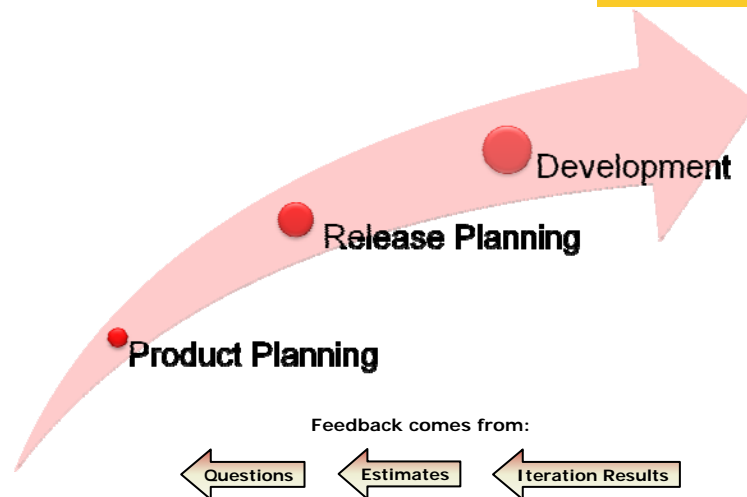

Hmm... What's in the Envelope?

Understanding of the capability (sub-capabilities)

Conditions of Satisfaction (exemplars)

The Value Proposition (relative importance)

The Cost & Risk (relative sizing)

Understanding of the implementation (options)

Hands-On

Refine a Couple of Stories

Stories Refined Incrementally

Development

Release Planning

Product Planning

Feedback comes from:

Questions | Estimates | Iteration Results

Capabilities & Stories Are Nested

"Epics" are larger stories. Still useful for visioning and planning, but not deliverable.

A customer can purchase the music in their cart.

A customer can pay for their music with a credit card.

A well-sized story must be cleanly and fully deliverable in a short cycle (< ½ iteration).

A customer can provide their credit card information.

Smaller stories are details. Usually they make up the understanding and elements of the conditions of satisfaction.

A customer can enter their card expiration date.

Epic Breakdown Hierarchy

On-Line Music Store

Music Catalog

Purchase Music

Itinerary Management

Music Browsing

Music Searching

Confirm Order

Pay for Order

Customer Reviews Prices

Customer Enters Payment

Customer Received Confirm

Themes

Saga

Epics

Stories
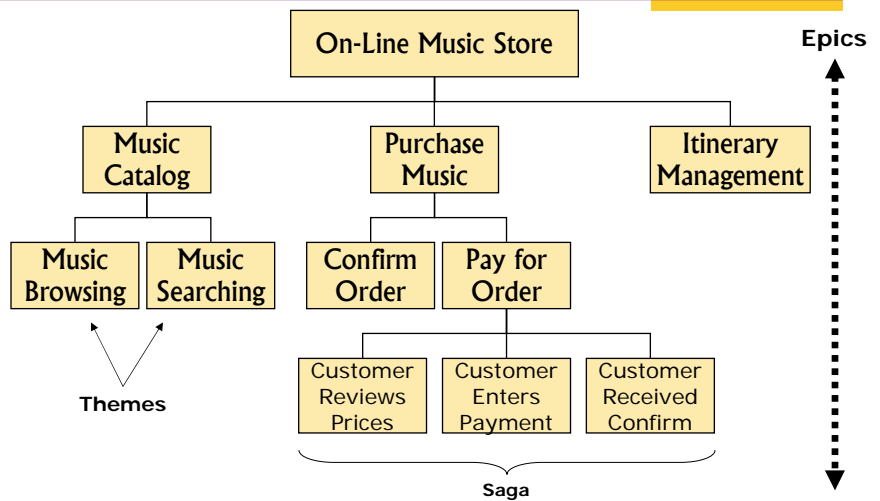
How Do We Break Epics Down?

- **By Value**
  - Essential vs. optional
- **By Sophistication**
  - Basic feature vs. bells & whistles
- **Incremental Build-Out**
  - Happy day to exceptional cases
  - Hard coded to automated
  - Fixed to configurable
  - Crude interface to full UI
  - Along the workflow (breadth first, then depth)
  - Split out research (spikes) & risk

Agile Logic
Deliver Better Software Faster

Example of Granular Story

> A Customer can pay for their music order with a credit card.
>
> - VISA card only.
> - Does not validate entered info yet.
> - Always approves, doesn't contact service yet.
> - Basic UI without graphics or ads.
>
> [Exemplar scenario to illustrate steps.]

Agile Logic
Deliver Better Software Faster

Hands-On

Break Down an Epic Story

Stories Broken Down Incrementally

Development

Release Planning

Product Planning

Feedback comes from:

Questions   Estimates   Iteration Results
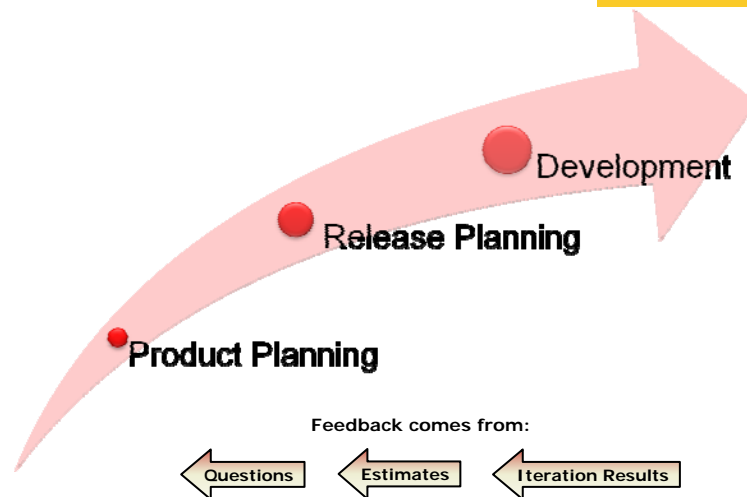
"INVEST" Criteria (Bill Wake 2003)

I   – Independent
N   – Negotiable (flexibility)
V   – Valuable
E   – Estimable
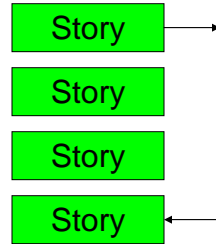S   – Small
T   – Testable

"Slicing" vs. "Layering"

User Interface

Business Services

Frameworks

Database

System Architecture
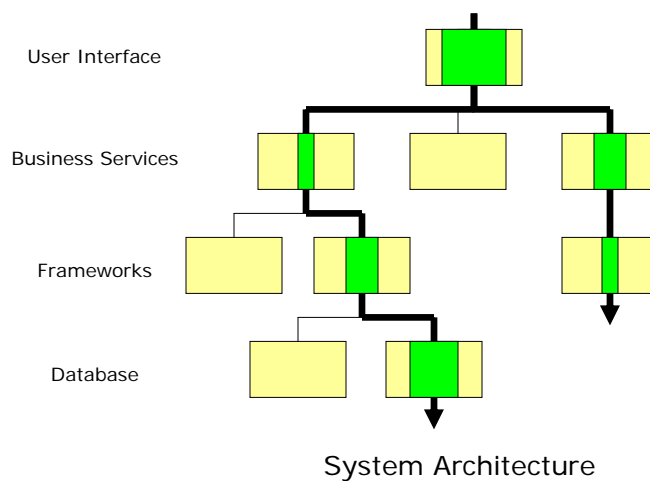
Story Dependencies

Stories should be as independent as possible.
But sometimes it's not entirely possible.

| Story |
| Story |
| Story |
| Story |

| Story |
| Story |
| Story |
| Story |
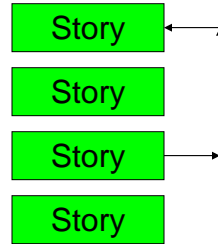
A healthy dependency
<u>goes with</u> the flow of value
delivery and priority.

An unhealthy dependency
<u>prevents</u> the delivery of
value in the desired priority.

Organizing Large Sets of Stories
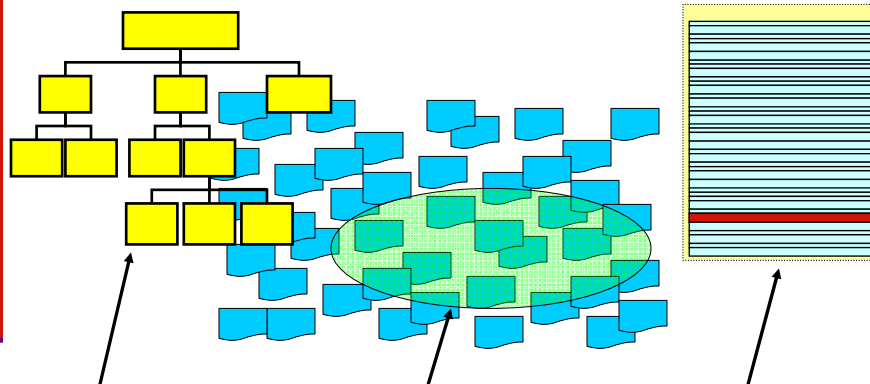
By Hierarchy
• Story Breakdown

By "Theme"
• Application Area
• By "Project" or Client
• Crosscutting (Security)
• Owner or Assigned Team

By Priority
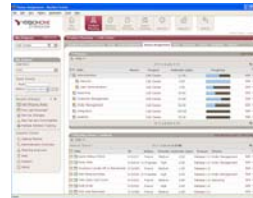• Value / ROI / Risk
• Dependencies

Formality and Tools

Start simple and keep it simple!

Note Cards Corkboards	Spreadsheets SharePoint	Application

- ✓ Make sure you understand what you need <u>first</u>.
- ✓ Find tools that work <u>with</u> your process.
- ✓ Make sure the tool <u>helps facilitate collaboration</u>.

Questions?

Thank You For Attending!

Paul Hodgetts
Agile Logic
www.agilelogic.com
phodgetts@agilelogic.com
(714) 577-5795

Agile Logic
Deliver Better Software Faster