

Virtually Agile

Astro Sabre (Matt Ganis)
<http://webpage.pace.edu/mganis>

IBM, Senior Technical Staff Member
Hawthorne, NY - September 20, 2007



Who am I ?

My Name is Matt Ganis

I lead a group inside IBM called Agile@IBM that consists of developers, testers, architects and project managers interested in pursuing Agile methods

Officially I work in the ibm.com organization as part of the site architecture team for the external website.

I'm also the lead architect for the ibm.com Virtual Branch Office in Second Life



“Internet Time”

“The Conventional wisdom about competition in the age of the Internet is that the business world has become incredibly fast and unpredictable, and we need to throw out the old rules of the game.....”

*For companies competing in the new information economy, the Internet is forcing managers and employees to experiment, invent, plan, and change their ideas **constantly** while they try to build complex new products and technologies”*

**Competing on Internet Time: Lessons from Netscape
and it's battle with Microsoft**
- Michael Cusumano and David Yoffie

What is Agile¹

Agile is an iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner with "just enough" ceremony that produces high quality software which meets the changing needs of its stakeholders.

¹ <http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm>



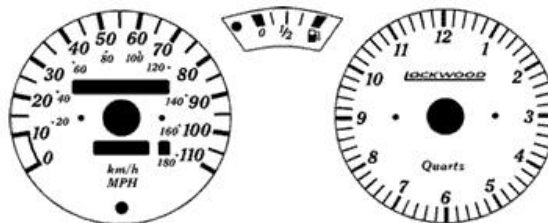
Why Agile ?

- **Many clients are struggling with application delivery issues:**
 - Poor IT relationship with the business
 - Track record of poor project delivery
 - Inability to deliver on-time, on-budget
 - Inability to delivery solutions that meet the needs of the business
 - Poor results with offshore delivery, or seek to avoid offshore delivery
 - Large project backlog
- **Internal and external IBM delivery projects are interested in Agile techniques to help address delivery excellence challenges**
 - Requirements often are not well-defined
 - Speed time to deliver critical business functionality
 - Reduce technical risk for first of a kind solutions

Agile Practices

XP is extreme in the sense that it takes 12 well-known software development "best practices" to an extreme

- Planning Game
- Small Releases
- Simple Design
- Continuous Testing
- Refactoring
- 40-hour work week
- Pair Programming
- Collective code ownership
- Continuous Integration
- On-Site customer
- Coding standards



Key Agile Terms

Term	Definition
Stories	A project conducted under an Agile Method is broken up into a set of very small deliverables called stories.
Velocity	Velocity is a method for measuring the rate at which teams consistently deliver business value in a software system (at what rate can they deliver stories)
Iteration	Software developed during one unit of time is referred to as an iteration, which may last from one to four weeks. Each iteration is an entire software project: including planning, requirements analysis, design, coding, testing, and documentation. Stories are implemented within iterations
Customer	The stakeholder that is responsible (i.e., has money) and “owns” the requirement

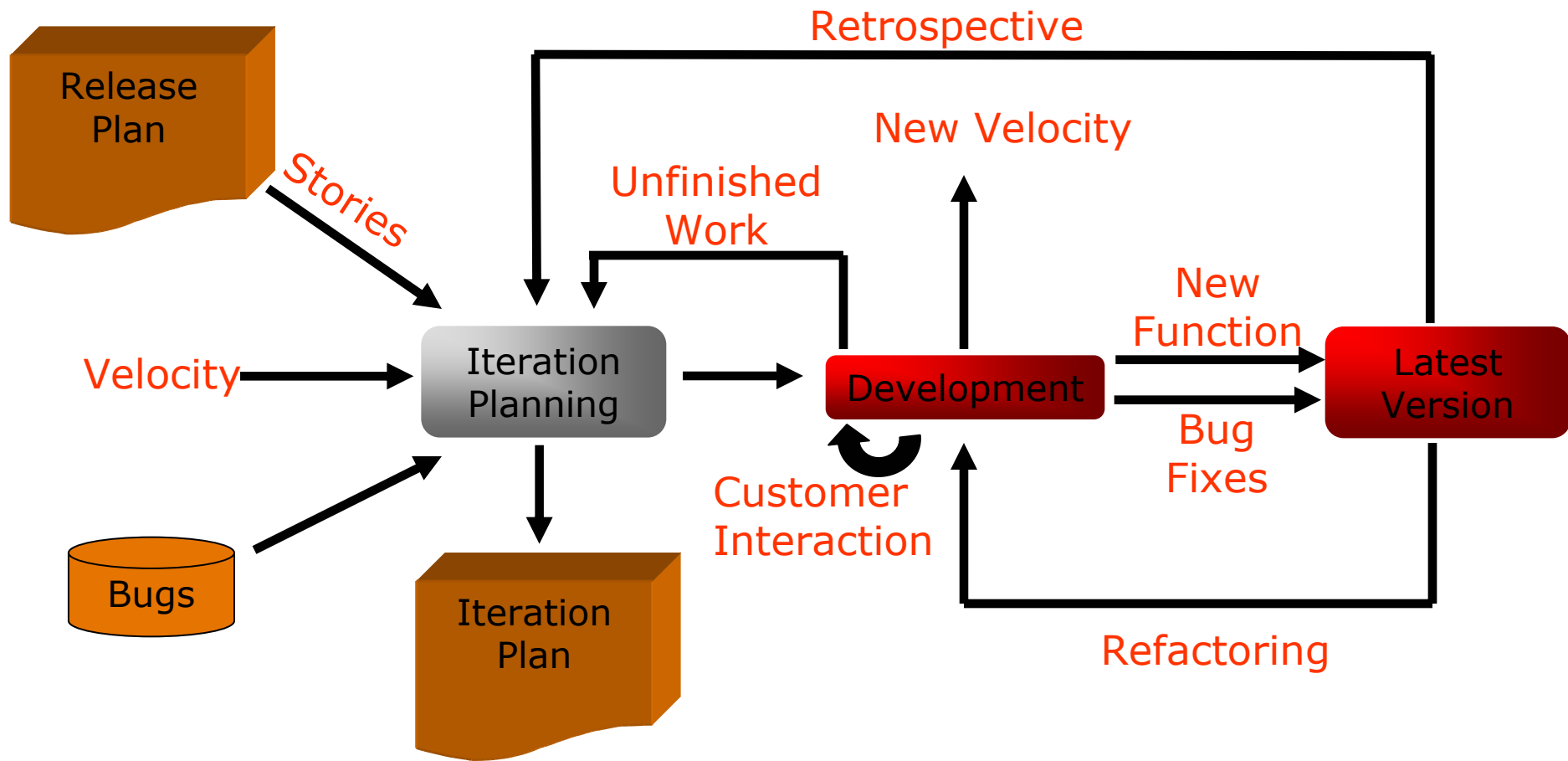
What makes a “good” story

(from Bill Wake: <http://www.xp123.com/xplor/xp0308/index.shtml>)

I N V E S T

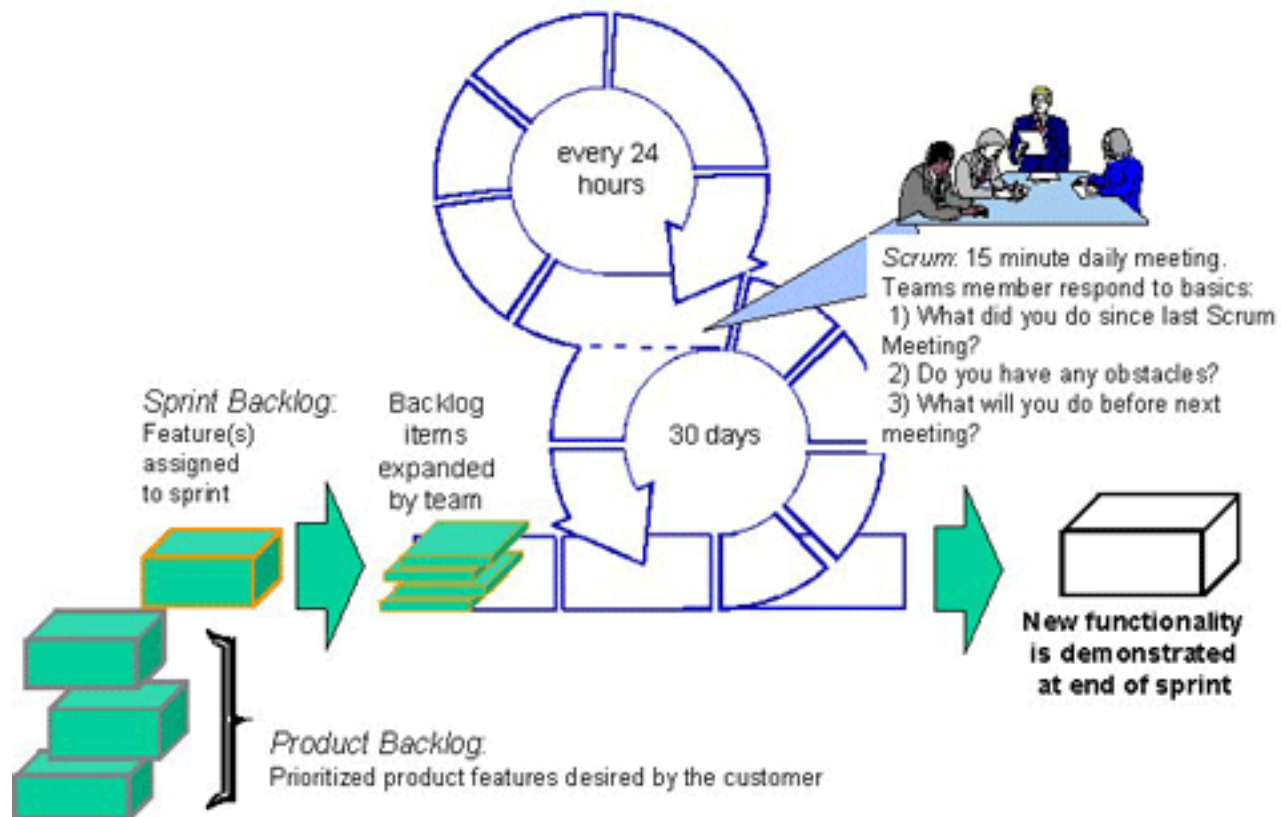
I	<u>Independent.</u> We can build them in any order. This allows us to select stories with the highest value
N	<u>Negotiable</u> - Remember that agile methods are typically variable in scope. That is the time line is fixed (iteration length) and the quality and scope are varied.
V	<u>Value.</u> Stories need to have real business value to the stakeholders so they can chose the most important” ones
E	<u>Estimate.</u> If a story can't be estimated (how long to complete) then the customer can't derive value or assign a priority to it.
S	<u>Small.</u> Having small stories is a result of estimable and negotiable. The larger the story the harder it is to estimate, the less flexibility in negotiation.
T	<u>Testable.</u> Stories need to be testable, otherwise how do you know the story is complete?

Iterative Development Flow



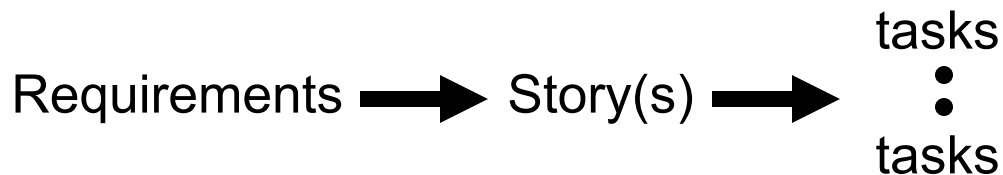
Scrum

(project management)



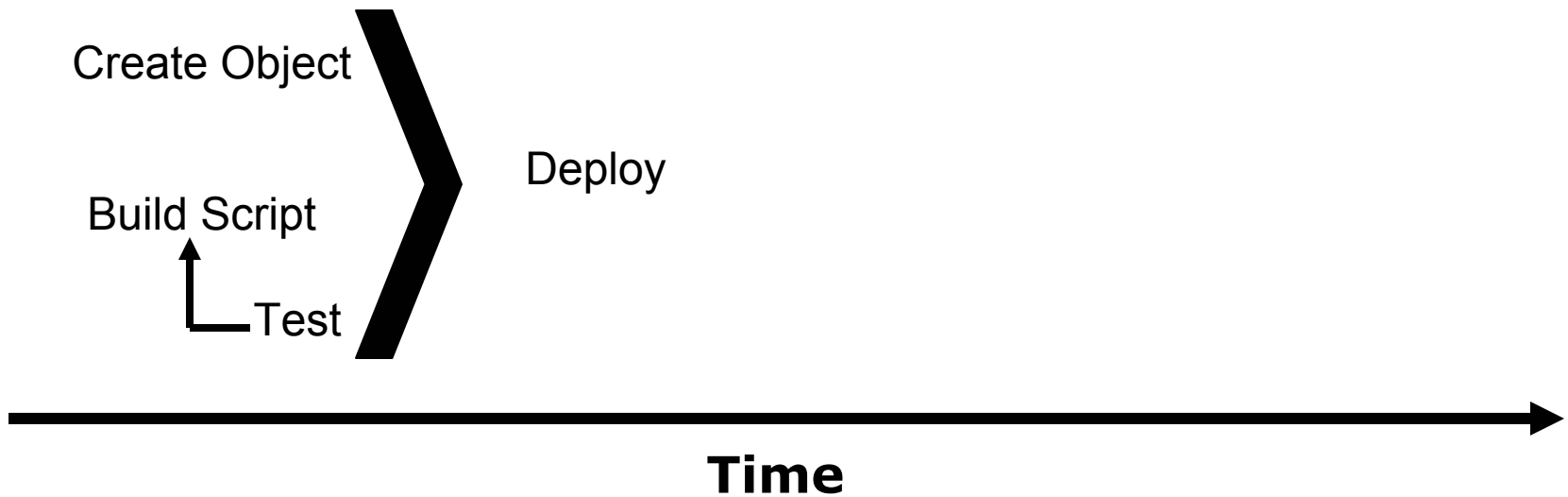
Second Life Object as stories

Objects are perfect examples for Agile stories. They have a value based on the function they include, can be independent and are negotiable



Sequential vs. Agile (incremental)

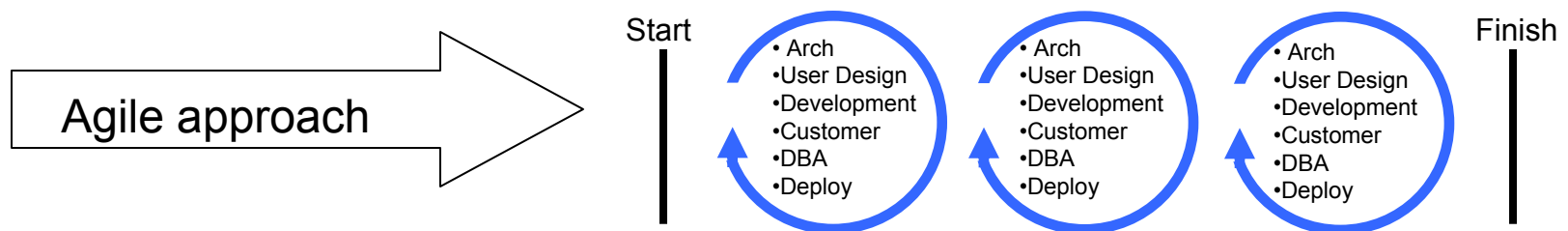
Create Object → Build Script → Test → Deploy



Story Examples

The Requirement:

We want a billboard (that can exist anywhere in Second Life) that will read a list of events from our calendaring system for a given month and display them in an effort to promote attendance. The calendaring system exists inside IBM (behind a firewall)



The Stories

(the requirement decomposes into at least 3 stories)

1. Create a billboard that can display “text on a prim” (hopefully sentences)
2. Read the text from a standard Notecard that can be dropped on the object
3. Add HTTP access to the billboard to retrieve a standard text file for all copies of the billboard

Benefits

- Value is delivered faster
 - We can also achieve a certain level of *cost avoidance*
- Managing rapid change of requirements based on business value
- Developers are learning and training each other
 - Stories will get more complex as skills grow