User Stories in Agile Development— A *Different Sort* of Requirements Artifact!

Bob Galen

President & Principal Consultant,

RGCG, LLC – Leading you down the path of agility…

www.rgalen.com bob@rgalen.com

Thank you to our sponsors

Pragmatic Marketing®

brainshark™

enthiosys

Creative

Graphic Design
Digital Prepress
Specialty Advertising
Creative Marketing
Print Media

Lance L. Bush

919.577.0036
Cell: 919.753.5436
518 Powhatan Drive
Fuquay-Varina, NC 27526
ArtHouse518@nc.rr.com

Cambria Suites®

Babymoon Cafe

DESIGN & INNOVATION
STUDIO HAGLER

SEQUENT®
LEARNING NETWORKS

iContact
Email Marketing Simplified™

Montie Design

montie roland
800-722-7987
montie@montie.com

MONTIE.COM

DEDICATED TO Excellence

VISION
RAPID RESPONSIVENESS
LEADERSHIP EXPERIENCE
SUSTAINABILITY

Allscripts™

GXS

TRIANGLE
INTERACTIVE
MARKETING ASSOCIATION

Introduction
Bob Galen

- Somewhere 'north' of 20 years experience ☺
- Various lifecycles – Waterfall variants, RUP, Agile, Chaos, etc.
- Various domains – SaaS, Medical, Financial, Computer Systems, and Telecommunications
- Developer first, then Project Management / Leadership, then Testing
- 'Pieces' of Scrum in late 90's; before Agile was Agile
- Agility @ Lucent in 2000 – 2001 using Extreme Programming
- Formally using Scrum since 2000
- Most recently at ChannelAdvisor as Dir. Product Development and Agile Architect/Coach (2007-2008)
- Connect w/ me via LinkedIn if you wish…

Bias Disclaimer:
Agile is THE BEST Methodology for Software Development…
However, NOT a Silver Bullet!

Outline

Introduction

1. User Stories
2. User Roles
3. Misc. (Challenges, Role of BA)
4. Workshop Close

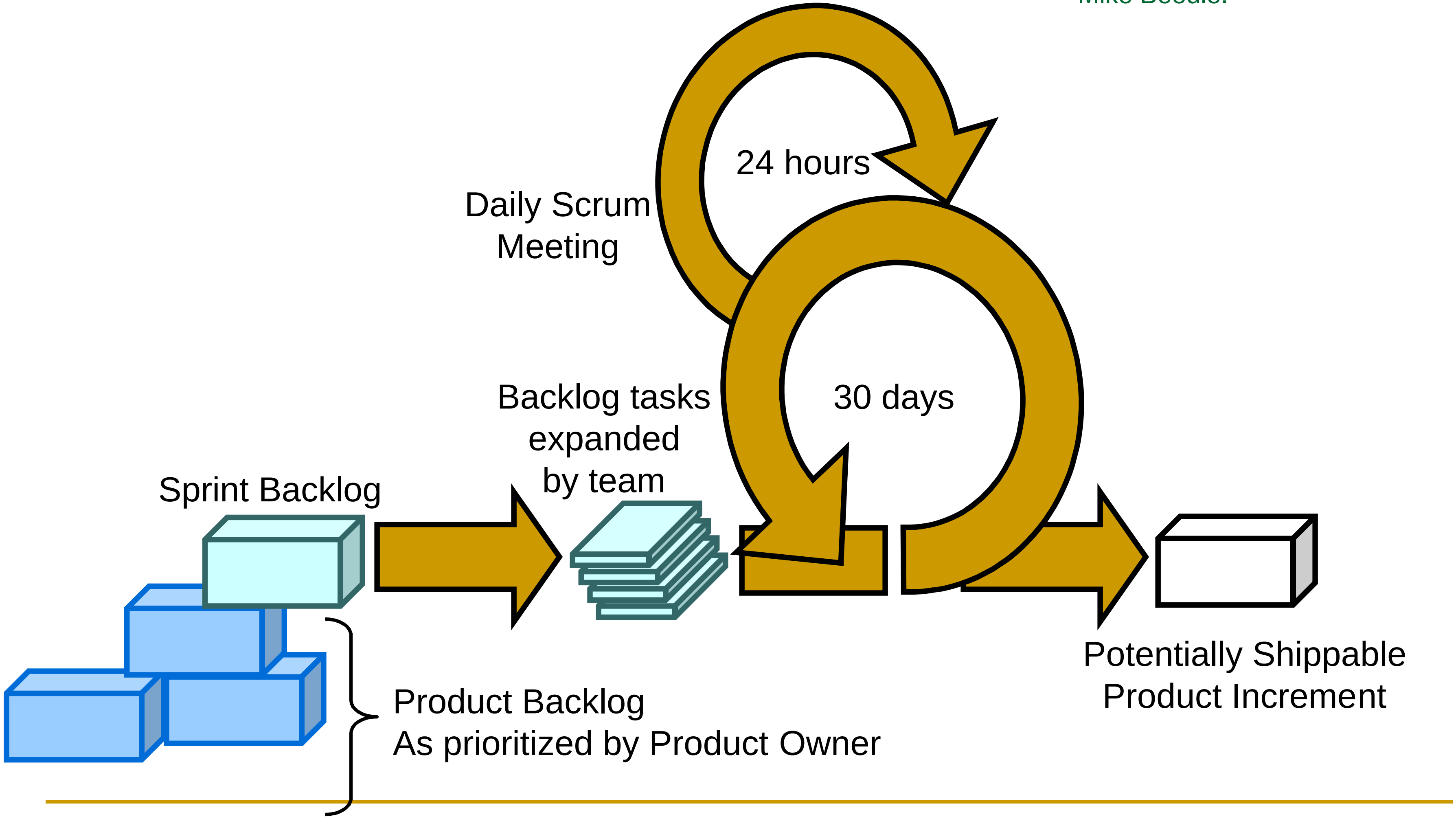
Scrum

We'll be using Scrum as the Agile reference framework for the class

- Agile Project Management Framework
- Roles:
  - Scrum Master, Product Owner & Team
- Sprint – iteration (15 – 30 days)
- Product Backlog – a prioritized list of: requirements, work to do, user stories
- Sprint Planning – mapping PB to tasks
- Daily stand-up (15 minutes, Pigs & Chickens)
- Sprint Review (demo) & Retrospective

Agility Example – Scrum

Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.


24 hours
Daily Scrum Meeting
30 days
Backlog tasks expanded by team
Sprint Backlog
Product Backlog
As prioritized by Product Owner
Potentially Shippable Product Increment

Basic User Stories

Traditional Waterfall Requirements

- Upfront definition
  - Established in the beginning often in great detail
- Signed off with business stakeholder and/or customer
  - Contractual in nature
- Change is considered “bad”
  - Usually change control mechanisms are instituted
- Scope confirmed at the end

- Requirement “issues” are often listed in the Top 3 failure factors within software projects
  - And typical statistics reflect between a 50 – 65% failure rate amongst cost, requirement and delivery time targets

Agile Requirements

- High level requirements (Backlog) established early
  - Established in the beginning often ambiguous
- Ongoing collaboration with business the stakeholder and/or customer
  - Effectively a “partnership” arrangement
- Focus on prioritization seeks to minimize gold-plating
  - While making cost / value and progress / capacity visible
- Change is embraced
  - With the cost of change being clearly visible throughout
- Scope confirmed & accepted iteratively

Traditional vs. Agile Roles

Traditional roles

- Customers are interviewed
- Architects and BA's write requirements
- Collaborative review & change
- Sign-off

- Key drivers: Team leads, BA's, Architects

Agile roles

- Team conducts story workshop to define the backlog
- Product Owner + Team prioritizes (orders) and clarifies
- Look ahead only a few sprints
- Continuous exposure & adjustment with discovery

- Key drivers: Testers and PO's drive the requirements within each sprint

User Stories

- Notion of Stories derived from Extreme Programming – XP
- Stories are requirements / features / units of work
- They are similar to use cases (as good a comparison as any)
- Written on a 3x5 card or Post-it note
- Capture a succinct item of work
- Are collaboratively defined by customer w/the team
- Are prioritized by and delivered to the customer

User Story Construct

As a <*role*>

I want <*system behavior*>

So that I realize <*some business value*>

User Story Construct

As a <*role*>

I want <*system behavior*>

So that I realize <*some business value*>

**And can see that it does <example>

User Story Examples

As a dog owner, I want to sign-up for a kennel reservation over Christmas so that I get a confirmed spot

As a dog owner, I want to check on the status of my dog (meals, walks, medication) so that I don’t worry

As a dog owner, I want to pay for my dogs kennel visit before or after they enter the kennel

User Stories
Parts of

- Front of card – describes the Story characteristics – flexible format
  - ❑ 1-2 sentences; short and succinct
  - ❑ I’ve seen limits of 5 words per Story
- Back of card – describes User/Customer Acceptance Tests
  - ❑ Mini “*contract*” to determine “*done*” or completeness
  - ❑ Combination of the two, effectively describes the story functionality
- Intentionally incomplete!

User Story Examples

As a dog owner, I want to sign-up for a kennel reservation over Christmas so that I get a confirmed spot

Verify individual as a registered pet owner
Verify that preferred members get 15% discount on basic service
Verify that preferred members get 25% discount on extended services and reservation priority over other members
Verify that past Christmas customers get reservation priority
Verify that declines get email with discount coupon for future services

User Stories
Parts of

- Third, most important part of the story is a *promise to have a conversation*
  - Right before & during implementation
  - Amongst the team members implementing the Story / Feature
    - Developer, Product Owner, Tester, Architect, etc.
  - Will drive (just in time, just enough) requirement artifacts
    - I need a wireframe to help with design & layout
    - Let’s develop a Use Case to elaborate additional functional details
    - We need more Acceptance Test coverage to drive clear business logic
    - Let’s put this on the wiki for traceability

Ron Jeffries
3 C's

- **CARD**
  - *Use a 3x5 card or other low fidelity means to capture requirements*

- **CONVERSATION**
  - *Certainly don't try and capture all detail on the card. Instead, sit down as a small implementation team (developer, tester, customer) and discuss the implementation from each perspective. Write a little code, run it, test it, and discuss it—then do it again!*

- **CONFIRMATION**
  - *Working with the customer, define a set of acceptance tests for each story*

Ron Jeffries
3 C’s

- What’s the most important ‘C’?
- I’d opt for the ongoing collaborative *conversations*
- What do you think is next in importance?
- I’d opt for the *confirmation* tests. They help illustrate design decisions in the code and promote sustained quality

Story Attributes: INVEST

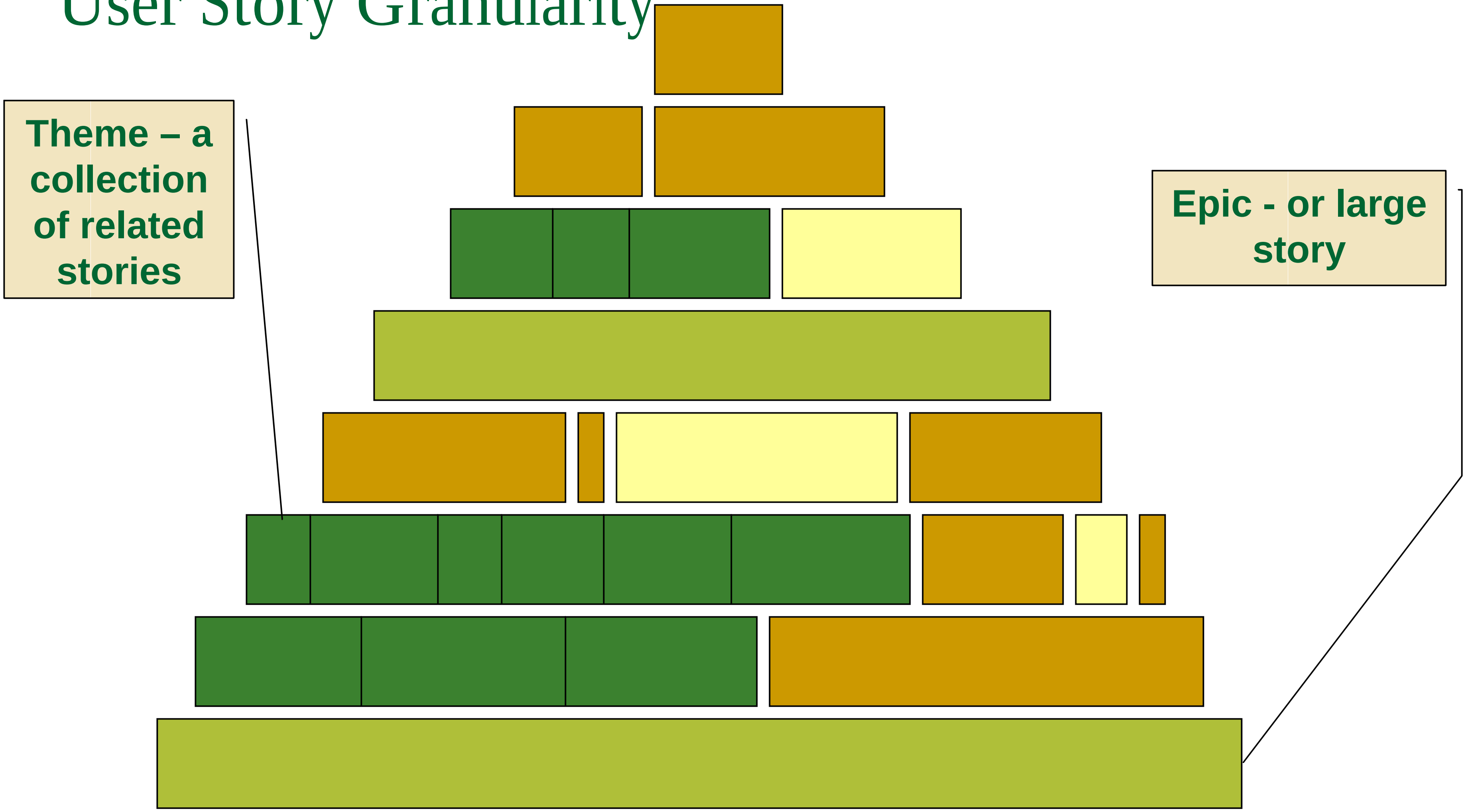
- Independent
  - as much as possible stories need to stand alone, with few dependencies to other stories
- Negotiable
  - they are not fixed “contracts”
- Valuable
  - prioritized so that the value to the customer is clear (relative to other stories)
- Estimatable
  - clear enough, with knowledge surrounding domain & technical points
- Small
  - fit easily within an iteration, requiring 1 to a few resources to implement
- Testable
  - usually expressed with acceptance criteria and/or automated tests

User Stories
Other Types

- A Spike is a story focused toward technical (or other feasibility) investigation
  - Result is information or knowledge; not production code
  - Typically leads to other development focused story(s) and improves the estimate(s)

- Sub-stories
  - At time we defined stories with too fine a level of granularity. This dilutes value delivery.
  - They should be aggregated into a larger Story for implementation & delivery

Product Backlog
User Story Granularity



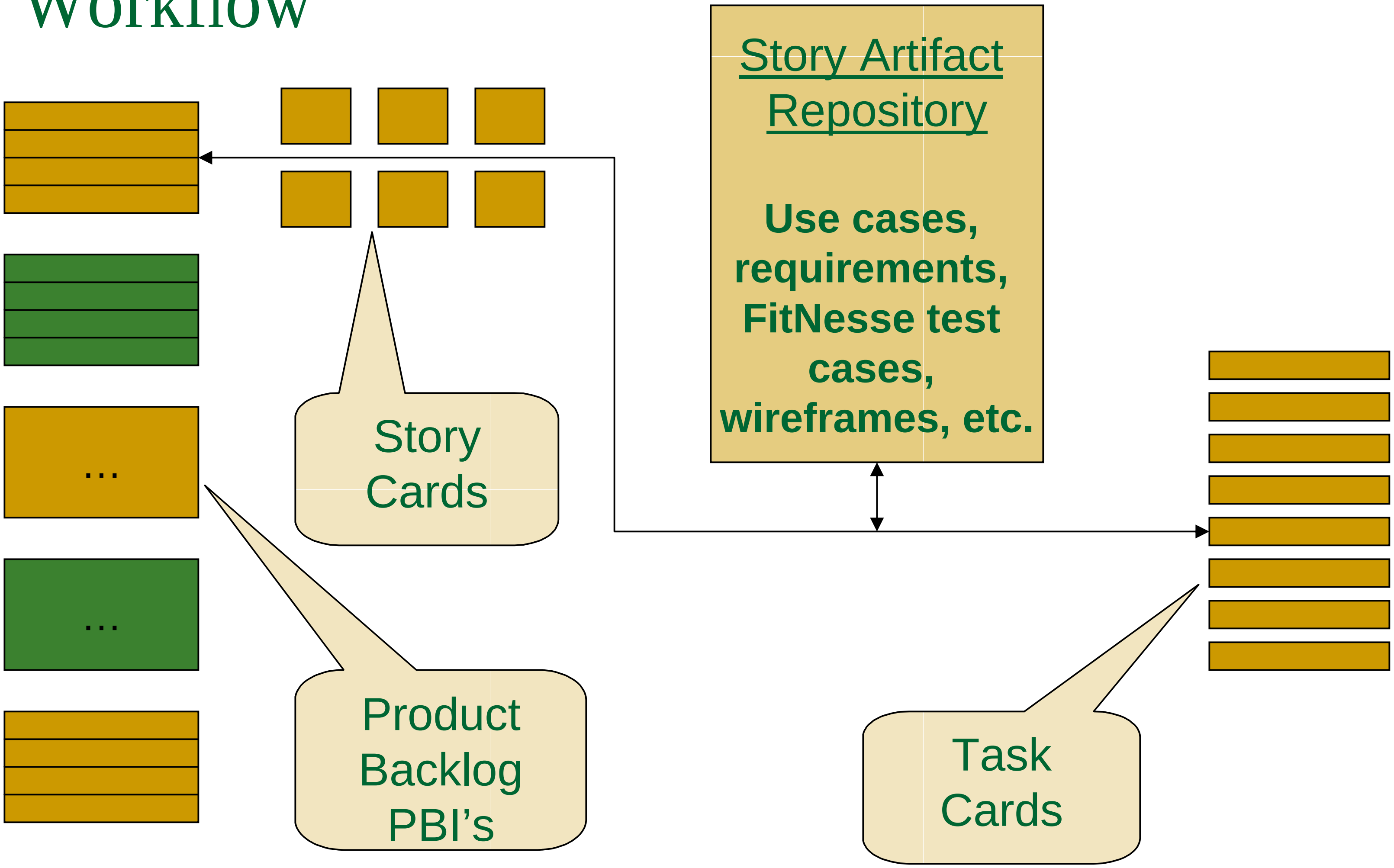
Splitting Stories

- Along operational or data boundaries

As a dog owner I can manage my pets reservations

- As a dog owner, I can reserve a kennel for vacation or holiday travel
- As a dog owner, I can cancel a reservation
- As a dog owner, I can reserve a kennel for my sick pet and order
- As a resident, I can reserve time to pick-up my pet
- As a dog owner, I can create my owner profile
- As a dog owner, I can update my owner profile
- As a vet, I can check on my pets medical conditions and update their

Product Backlog – to – Task Workflow



Story Artifact Repository

Use cases, requirements, FitNesse test cases, wireframes, etc.

…

…

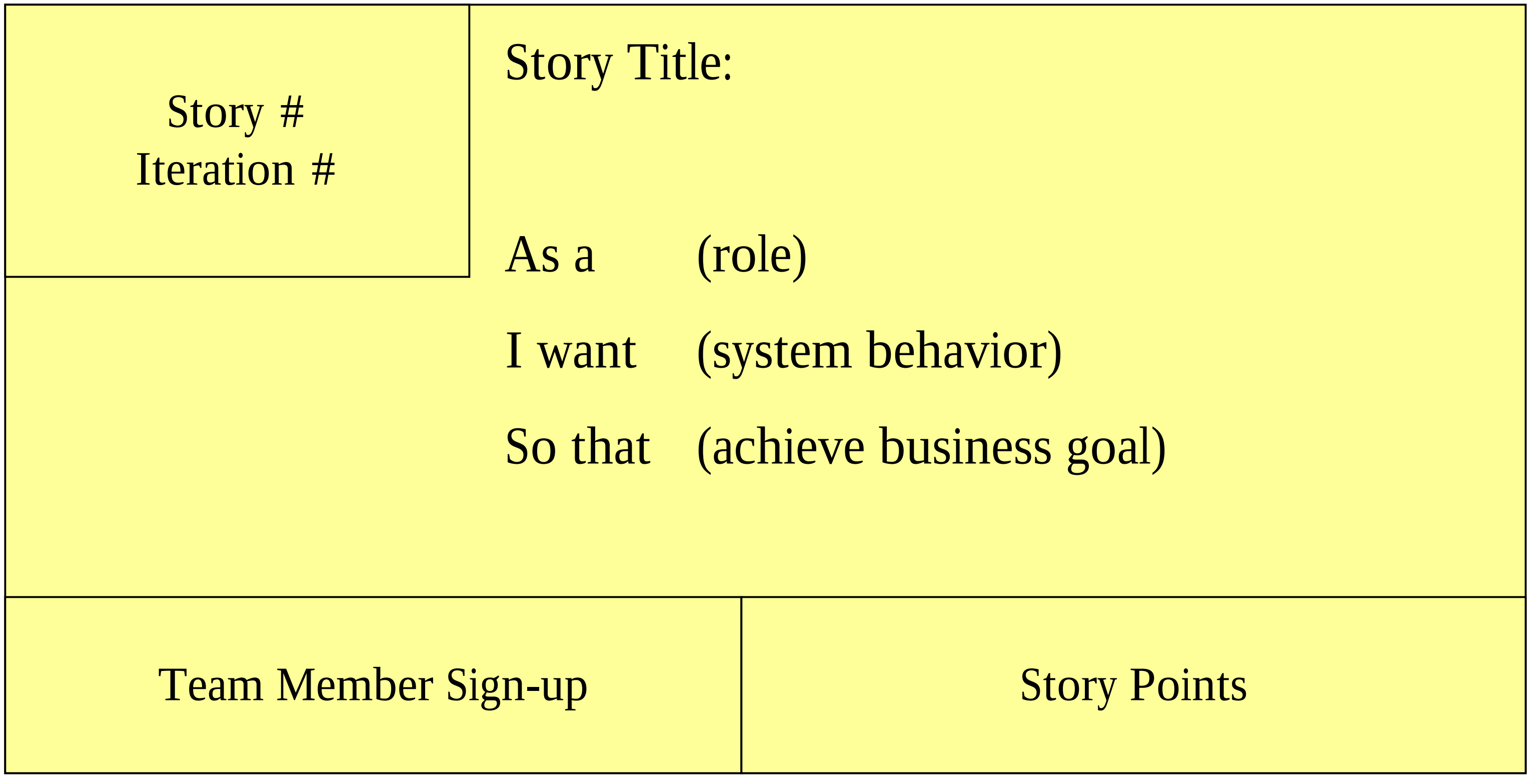
Story Cards

Product Backlog PBI's

Task Cards

User Story
Another Sample Format

Sample XP Story Note Format



Story #
Iteration #

Story Title:

As a (role)

I want (system behavior)

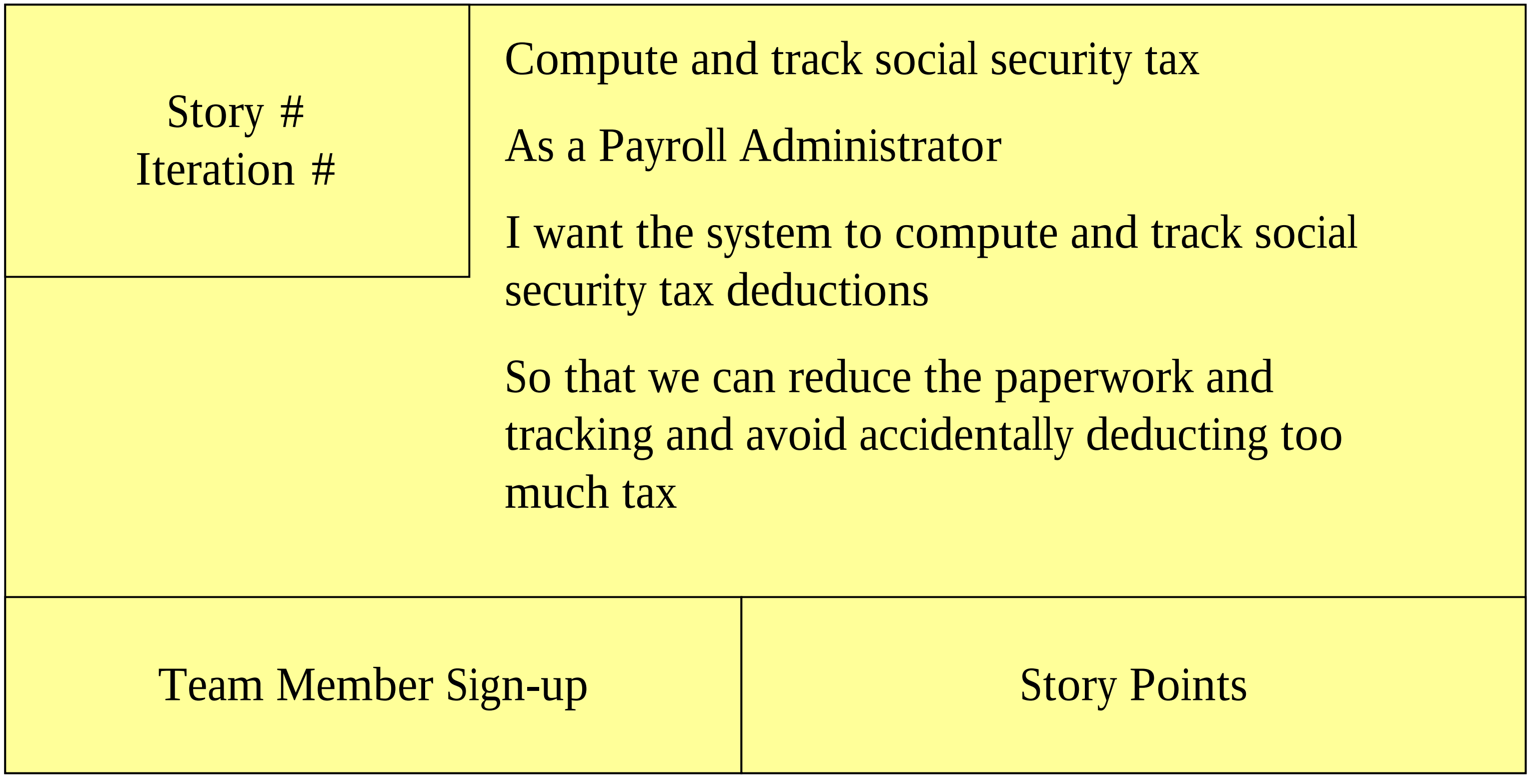
So that (achieve business goal)

Team Member Sign-up

Story Points

User Story
Another Sample Format

Sample XP Story Note Format



Story #
Iteration #

Compute and track social security tax

As a Payroll Administrator

I want the system to compute and track social security tax deductions

So that we can reduce the paperwork and tracking and avoid accidentally deducting too much tax

Team Member Sign-up

Story Points

User Story
Another Sample Format

Sample XP Story Note Format

Compute and track social security tax

Acceptance Tests

I do this	This happens
-Pay at or above $1,000	Deduct 1.3% for social security
- Previous total at limit	Deduct $0
- Near annual limit	Deduct 1.3% minus limit

User Roles

Identifying the User

- Often we overload many different types of users as a single “The user” in our requirement definition
- Assume that all users have the same goals
- Write too many of the stories from the same “generic user” perspective

User Roles

- Broaden our view to the needs / goals for the software

- Users
  - How they will use the software
  - What they will use it for
  - The perspective or background for each user
  - Qualify what’s important for each user
  - Identify their level of competency or familiarity with the software

- Based somewhat in *Usage-Centered Design* from Constantine & Lockwood

Role Examples

- The *Kennel Owner* who wants to monitor animal reports showing capacity and revenue on a daily basis
- The *New Pet Owner* who is using the kennel for the first time and is looking for assurance on overall quality
- The *National Kennel Evaluation Service* who is going to list the kennel and specific size and quality attributes in a national register
- The *Frequent Customer* looking for special treatment for their pet over the holidays

User Role Brainstorming
Steps

1. Engage a group for a brainstorming meeting with cards
2. Attendees should include team members, customers, stakeholder, potential users
3. Write individual roles on cards
4. Can yell out the role as you place the card on the table
   1. Or a facilitator can do that as part of organizations

This is a quick exercise, so 10-15 minutes for data initial role collection

User Role Brainstorming Types

- When brainstorming, consider true roles:
  - Power users
  - Simple users
  - Important or valuable users
  - Malicious users
  - Test users
- Also consider roles that are not people
  - Interfaces or other systems
  - Organizations or companies
  - Regulatory agencies

User Role Brainstorming Consolidation

- As you develop a rich set of roles, start to think about organizing & consolidating them

- For example, *Kennel Customer* might include:
  - First-time customer
  - Repeat Customer
  - Priority Customer
  - Returning Customer

- Or *Administrative User* might include:
  - The Owner, Vet, General Manager, Team Leaders, Employees

User Roles
Advantages

- Start incorporating your roles into your Stories
  - Helps with developing a broader set of Stories
- Users become much more tangible
  - We can step into their role and better envision their needs
- Avoid the tendency to have a generic *User* and then overload their role
  - Speaking in terms of “The Repeat Customer” or “The Owner” when crafting Stories
- Accommodates and clarifies technical and system roles

User Roles
Special Roles

- Systems, Services, API's
  - As our *Payroll Provider*, I need to access 401K information for end of year processing
- Specialized team members
  - As a *Customer Account Manager*, I need to reset these system when the queue processing is "stuck"
- Security risks/hackers and other abusers of the system
  - Define Stories that would prevent these potential attacks

User Roles
Special Roles

- Personas
  - Are further elaborated roles – usually formally documented
  - Perhaps contain
    - Name
    - Picture
    - Education background
    - Level of experience
    - Motivation or goals
    - Job or position activities

- Help to further envision and *place a face* on the user and add *depth to our envisioning*

Misc. Topics

User Story Ordering Prioritization

- From a Product Backlog perspective, User Stories need to be ordered by priority

- Instead of working with individual Stories, try to organize related Stories into Themes. This is primarily the domain of the Product Owner
  - Organize needs into Themes, Pack Stories into Themes, and Prioritize Themes

- Usually Themes can be driven out of the Story writing workshop
  - When organizing and grouping your stories

Themes
Typical Workflow

Early Themes:
Environment Setup, Project Feasibility,
Architectural Models, Team Formation

Middle Themes:
Feature Acceleration, Product Maturation
Architecture Stabilization

Later Themes:
Integration & ‘Ility Testing, Customer Readiness
Deployment, Next Release

Typical Challenges for User Story Adoption

- Creating the initial Story / Backlog
  - Just do it; only need the ‘tip’ well defined
  - Rarely do you get to “everything”
- Granularity—too large vs. too small
  - Changes as Stories move closer to execution
  - Getting the team to participate in refining estimates
- Getting stuck on developing perfect Stories
  - Team naturally get better at defining them over time; Each teams’ view is somewhat unique
- Making the Backlog Transparent
  - Placed on a wiki; generally widely available for view
  - Encourage changes; additions, reordering!

Typical Challenges for User Story Adoption

- Team members contributing
  - Everyone can contribute (write) a User Story and add it to the Backlog – Operations, Technical Writing, Testing, Non-functional Requirements
- Contributing - Acceptance Tests
  - They're not just tests! And not just for Testers to write.
  - Help refine the design; help define 'Done'; help focus prioritized, functional testing
- Viewing Stories as 'static'
  - Stories on Product Backlogs need to be Organic
- Forgetting to TALK
  - Trying to have them replace conversation OR trying to capture all discussion ON the Stories

Business Analyst Role in Agile Development

- You are a Team Member!
  - All work is performed within the context of your Scrum Team
  - Your efforts are tasked as part of Sprint Planning (task cards), achieve Done-Ness, and demonstrated in the Sprint Review
- Partner with the Product Owner / Customer
  - Rarely should the BA “Be” the Product Owner; think customer facing, Product Management
- Partner with Testing
  - Developing & refining Acceptance Tests
  - Automating them as part of Sprint activity; for example in FitNesse
  - Support the notion of ***Executable Requirements***

Business Analyst Role in Agile Development

- User Story focused; Less Product Backlog focused
  - Scribing / writing them
  - Helping with Acceptance Tests
  - Grouping, constructing themes
- Facilitate Story Writing Workshops
- Continuously refine / ‘Groom’ the Backlog with the team
  - Product Owner leads this effort – workflow, milestones, and goal setting
  - Estimates belong to the team
- Work ahead – anticipate needs
  - Architecture, usability, interface design, etc.
  - Perhaps ‘skewed’ work within Sprints

Workshop Exercise

Simulation
Background, Charter, Goals

- We're going to build an addition off of your house as an exercise
  - It will be a 20'x30' extension
  - It will not have a basement, but a simple foundation
  - Ceilings will be vaulted – slanting from 10 – 20 ft.
  - The wall facing your garden will be all windows
  - You're adding a ½ bath to the corner of the extension
  - And there is a side section that will serve as a sitting / sun room
  - There are several ceiling fans and skylights
- You are the General Contractor and need to plan

Simulation Step #1
Story Creation

- Let’s brainstorm some User Story cards for building and decorating the room
- Concentrate on High Level requirements, needs, tasks, work, milestones, steps, etc.
- Simply get ideas on the cards

Simulation Step #2
Adding Roles

- Roles help to remind of us work to do – beyond simply the “User”
- For example:
  - As a rough Carpenter; as a finish Carpenter
  - As a Plumber, Dry waller, Painter
  - As a General Contractor
  - As a Home Owner

Example: As a Home Owner we need to pick out paint colors so that we effectively coordinate the new room.

Thank You!

Questions?

Thank you to our sponsors

Graphic Design
Digital Prepress
Specialty Advertising
Creative Marketing
Print Media

Lance L. Bush
919.577.0036
Cell: 919.753.5436
518 Powhatan Drive
Fuquay-Varina, NC 27526
ArtHouse518@nc.rr.com

montie roland
800-722-7987
montie@montie.com
MONTIE.COM
DEDICATED TO Excellence

Contact Info

Bob Galen
RGalen Consulting Group, L.L.C.

Agile focused training, coaching & consulting

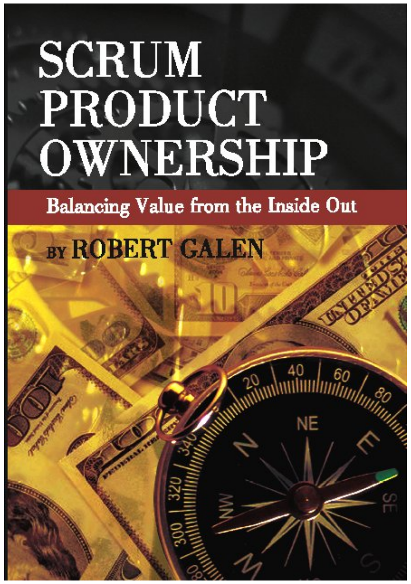
PO Box 865, Cary, NC 27512
919-272-0719

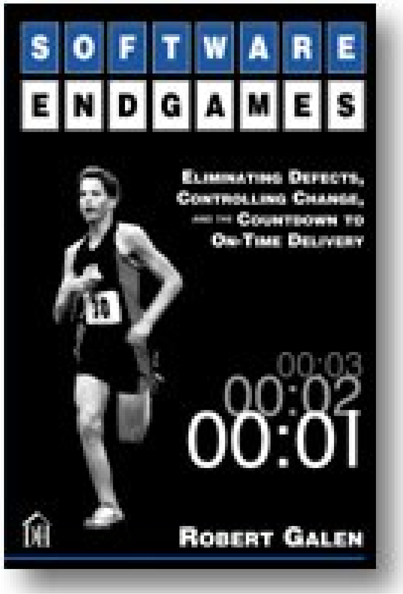
www.rgalen.com
bob@rgalen.com

Scrum Product Ownership – Balancing Value From the Inside Out published by RGCG in 2009.

Software Endgames: Eliminating Defects, Controlling Change, and the Countdown to On-Time Delivery published by Dorset House in 2005.

Go to www.rgalen.com for purchasing / for order info, misc. related presentations, and papers.




SOFTWARE
ENDGAMES
ELIMINATING DEFECTS, CONTROLLING CHANGE, AND THE COUNTDOWN TO ON-TIME DELIVERY
00:03
00:02
00:01
ROBERT GALEN