Kanban-based Scheduling Systems

Richard Turner
Stevens Institute rturner@stevens.edu

SEPG North America Conference
October 1, 2013
Pittsburgh, PA

www.SERCuarc.org
Agenda

- Introduction
- The Kanban Concept (On-demand scheduling)
- An Example from the Current Research
- Questions
The Systems Engineering Research Center

- The SERC is a US Department of Defense University-Affiliated Research Center (UARC)
- The networked national resource to further systems research and its impact on issues of national and global significance

The systems research and impact network
Introduction

It’s a dangerous business, Frodo, going out your door. You step into the Road, and if you don’t keep your feet, there is no knowing where you might be swept off to.

Gandalf the Wizard in J.R.R. Tolkien’s *Lord of the Rings*
The World is Changed

• Traditional project management (like systems engineering) is based on the assumption that work is predictable and stable
  — Requirements are stable
  — Resources and technologies are stable
  — Values remain stable
  — There is sufficient time to complete the work

• So 20th Century! In the 21st Century
  — System contexts have multiplied, and change in customer needs and developer solution technologies has accelerated.
  — Requirements are less tangible, more evolving, and sometimes emergent
  — Systems are both complex and constantly adapting

• PMI has adapted (finally)
  — V5 of the Guide to the PMBOK provides for both predictive and adaptive project lifecycles.
Problems and Symptoms

• Systems Engineering Disengaged from Software Engineering
  — Lack of responsiveness to product team requests
  — Late delivery of SE products needed by product teams
  — Product teams see SE as a barrier rather than an enabler

• Poor management visibility into work, relationships between products/requirements/architecture, impacts of change
  — IMS issues
  — Poor prediction of capability availability
  — Overloaded product teams

• Operational environment overwhelms traditional governance methods
  — Cowboy engineering
  — Unscalable or unsupportable capabilities delivered
Caution! Specific Target Environment for the SERC Research Under Way

• Systems engineering where rapid response software development projects incrementally evolve capabilities of existing systems or SOSs
  — C4ISR
  — Intelligence community
  — Hospital systems
  — Platform-based systems

• That does NOT, however, preclude it from being applicable outside that target; of course it doesn’t guarantee it
Predicted (Desired) Results

- Better *visibility* and *coordination* managing multiple concurrent development projects
- More effective *integration* and use of scarce systems engineering resources
- Increased project and enterprise *value* delivered earlier
- More *flexibility* while retaining *predictability*
- Less blocking of product team tasks waiting for SE response
- Lower governance overhead
Kanban Concept (On-demand scheduling)
A kanban (signal card) is part of lean manufacturing
- Created to the agreed capacity of the process; one card is associated with each piece of work
- Work can mean the
  o creation of a part
  o integration of a part into an assembly
- Once all cards have been associated, no more work in that process can begin until some piece of work is completed and the card becomes available
- Kanban uses visual signals to
  o synchronize the flow of work with process capacity
  o limit the waste of work interruption
  o minimize excess inventory or delay due to shortage
  o prevent unnecessary rework
  o provide a means of tracking work progress.
Kanban in Thoughtwork

• The components of production are ideas and information

• In software and systems, kanban systems have evolved into a means of smoothing flow by balancing work with resource capability and increasing visibility and communication

• The concept includes the limiting of work in progress (WIP) according to capacity

• In that way, it is characterized as an on-demand or “pull” system, since the work is pulled into the activity as capacity is available rather than “pushed” via a schedule.
Why Limit Work in Progress?

- Psychologists agree that multi-tasking comes at a productivity price: context switching.

- Weinberg calculated the number of concurrent tasks and productivity; Limiting WIP can maximize productivity.

- Limiting WIP also accelerates the delivery of value by completing work rather than simply accepting more. *(Don’t take on more work, finish something!)*

- Too tightly controlling WIP, however, can impact the ability to manage work according to value (liquidity).

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Time/task</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>40%</td>
<td>80%</td>
</tr>
<tr>
<td>3</td>
<td>20%</td>
<td>60%</td>
</tr>
<tr>
<td>4</td>
<td>10%</td>
<td>40%</td>
</tr>
<tr>
<td>5</td>
<td>5%</td>
<td>25%</td>
</tr>
</tbody>
</table>
• **Work Item**
  — The item controlled in the kanban system. A work item has a definition, a Class of Service, a value and often a rough estimate of work effort required. Value of a work item is determined by some value function, which can vary over time.

• **Effort Required**
  — The approximate size of work in person-units of time. May be a negotiated function of desired quality. Rarely accurate at first (ROM suffices). Work items should be reasonably homogenous within a kanban (same size, same resources, same value function).

• **Transit Time (Lead Time)**
  — The time measured from entrance of a particular work item into a Kanban to its delivery to the downstream customer.

• **Backlog**
  — A queue containing upstream customer work items awaiting service by a kanban system.

• **Work-In-Progress Limit**
  — Limit of work items allowed in progress at one time within an activity. Often initially set to twice the number of resources, but used to regulate and optimize flow and slack.
• **Activity**
  
  — Value-adding work that can be determined as complete. Includes: a set of resources, a WIP Limit and a done queue. Allows allocation of effort to complete a work item.

• **Done Queue**
  
  — A limited queue that holds work items awaiting processing by an activity. The items in the queue may be considered part of the Activity WIP or the queue may have a specific limit. The queue cannot be unbounded in order to maintain the kanban pull effect.

• **Resource**
  
  — An agent for accomplishing work; may be generic or have specialized expertise. May include specific productivity. Usually associated with a specific activity, but may be shared across activities. Resources can swarm to alleviate bottlenecks or handle certain Classes of Service.

• **Cadence (prioritization and delivery)**
  
  — The rhythm of the kanban system.
  
  — Prioritization cadence defines the planning horizon for the KSS.
  
  — Delivery cadence allows bundling work items if desired by the downstream customer.
  
  — Not necessarily an iteration. Kanban still allows for iterations but decouples prioritization and delivery to allow them to vary independently of cycle time according to customer desires, domain, and costs.
• Work Item Selection Policies
  — Rules for selecting the next work item from the backlog or a ready queue when an activity has less work than its WIP limit; depends on both Class of Service and Value Function, and leads to specific flow behaviors.

• Class of Service
  — Provides a variety of handling options for work items. May have a corresponding WIP limit for each activity to provide guaranteed access for work of that class of service. CoS WIP limit must be less than the activity’s overall WIP limit. Examples are expedite, date-certain and normal. CoS may be disruptive (such as expedite) and is the only way to suspend work in progress.

• Value Function
  — Estimates the current value of a work item within a CoS for use in the selection algorithm.
  — Most often based on cost of delay, but can be simple (the null value function would produce FIFO) or a complex, multiple kanban-system, multi-factor method considering shared scarce resources and multiple cost/risk factors.
  — The means of prioritizing work items. There may be multiple value functions that return independently established values for each hierarchical layer within the KSS. For example, in SE, the overall systemic value of a work item may differ from the one that the project-level value function would return.
• Visible Representation
  - A common, visual indication of work flow through the activities; Often a columnar display of activities and queues. May be manual or automated. Shows status of all work-in-progress, blocked work, WIP limits. It is a characteristic that provides transparency enabling better management. Difficult to model. Provides system wide understanding of status and value, and encourages collective responsibility for flow.

• Flow Metrics
  - Includes cumulative flow charting and average transit time.
Visual Representation: Kanban board shows flow and bottlenecks

**Kanban Limit** – regulates WIP at each stage in the process

**Flow** – from Engineering Ready to Release Ready

*From David Anderson*
Cumulative Flow Diagram Tracks WIP

WIP is directly related to Lead Time and Quality
A Generic Kanban-based Scheduling System

Upstream Customers
Work (Backlog)

Ready Queue
(Limit=6)

Activity
(WIP Limit=8, Resources=4)

Completed Work

Work Flow

- Work Item waiting for selection
- Normal Class of Service Work Item (NCOS)
- Special Class of Service Work Item (SCOS)
- Expedite Class of Service Work Item (ECOS)
- Resource (Individually numbered)

NCoS, WL=5

SCoS, WL=1 (included in activity WL)

ECoS, WL=1, (extends activity WL if necessary)
Examples of networked KSSs
Systems Engineering as a Service
• This section describes our thinking about how to successfully cast the SE process as a set of inter-related services

• We do not enter into this lightly

• We only touched on it in our first phase work

• Our latest funding includes investigating it, so we want to share our ideas with you and get your feedback and ideas
Rationale

• On-demand scheduling system for systems engineering
  — Focused on the rapid development environment
  — Attempts to merge the SE flow and the software development project flow
  — Not simply lay SE functions on top of project activities without concern for the rapid-response constraints
  — Maximize efficient use of scarce SE resources

• SE activities (work items) need to be defined and available for projects and the system owners to select for the ready queue

• The concept of services fits the need to encapsulate work, and provide a common value stream among project development personnel, SE, and the enterprise
Value/Priority for Servicing

- Maintaining prioritization across multiple stakeholders is resource-intensive and can cause delay

- Kanban forces stakeholders to agree only about what enters the kanban system queue next

- Stakeholders include customers/users, projects, executive management, and higher level systems engineering management
  - Negotiations with disparate levels of authority are difficult
  - Value functions consider inputs from all customers and help calculate value according to explicit policy
  - Service Level Agreements and Classes of Service help
An Example Implementation in a Healthcare System of Systems
Healthcare SoS

- Custom software SoS constituent systems include patient management, pharmacy, laboratory, radiology, and telemetry.
- Systems share a single database for all patients and personnel related to a given health care site.
- Interfaces to other health care systems are maintained.
  - Custom legacy systems, COTS products, and medical devices.
- The health care system’s primary goal is to
  - Support patient health care delivery.
  - Support coordination across a variety of health care providers.
- Key overarching requirements are to ensure patient-safety and to protect patient information.
Information/work Structure

MC 1

R 1
Product 1

R 2

R 3

R 4

R 5

R 6

MC 2

R 7

R 8

MC 3

Product 2

Product 3

Product 4
Proposed KSS Network Structure

Executive/Stakeholder Management (Customer)
- SLA establishment and monitoring
- Strategic planning
- Capability prioritization

Capability Engineering
- Analyze needs and alternatives
- Refine capabilities
- Develop requirements
- Allocate requirements
- Form cross organizational teams
- Cross-product and specialty engineering
- Validate and fully enable capabilities

Work Flow

Needs Backlog*

Product/Domain Engineering

Users
- Customer relations
- Initial Triage

Individual Product Team
- Product SE
- Identify SW Features
- Allocate features to SWDT
- Integrate features into requirements

SW Development Team

Pharmacy Domain Team

Network Domain Team

* All organizations can contribute to the Needs Backlog
## Classes of Service

<table>
<thead>
<tr>
<th>CoS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Critical</strong></td>
<td>Safety, security, or other emergency work items. <strong>Disruptive:</strong> requires necessary resources to stop current work and complete it.</td>
</tr>
<tr>
<td><strong>Expedite</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>Very high priority work items such that this work takes priority over other work in the ready queue. Not Disruptive.</td>
</tr>
<tr>
<td><strong>Date Certain</strong></td>
<td>Work items that must be completed by a specific date or there will be significant consequences.</td>
</tr>
<tr>
<td><strong>Standard</strong></td>
<td>The normal CoS for the development organizations work.</td>
</tr>
<tr>
<td><strong>Background</strong></td>
<td>Work that must go on but is usually not time critical. It includes things like architectural enhancements, low-level technical debt, or research and environmental scanning</td>
</tr>
</tbody>
</table>
Executive/Stakeholder Management  KSS

- Determines which proposed capabilities (or enhancements) are approved to develop
- Assesses the value of the capability against its expected cost and schedule to develop.
- Tracks status by development state of approved but “not fully deployed” capabilities – WIP
- Informs decisions on organizational strategy, resource staffing, and funding priorities.
# Executive/Stakeholder Management Dashboard

## ESM Dashboard

<table>
<thead>
<tr>
<th>Capabilities in Progress</th>
<th>CoS</th>
<th>Value</th>
<th>Total # of Requirements</th>
<th># Requirements Completed</th>
<th>% Value completed</th>
<th># Requirements in Progress</th>
<th>% Value in Progress</th>
<th>% Requirements with work items blocked</th>
<th>Expected Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 4 (CRITICAL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## ESM Backlog

<table>
<thead>
<tr>
<th>Capabilities in Progress</th>
<th>Items in backlog</th>
<th>CoS</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability 13</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Capabilities (% complete)

- Capability 4 (CRITICAL)
- Capability 5
- Capability 3
- Capability 2
- Capability 1

## Total Value in Progress (completed)

- Last Three Months
• Represents all capability SE activities, including specialty SE support for PTs

• Creates capability descriptions incorporating needs identified/prioritized by ESM.

• Balances various SE resources (internal activities and cross-organizational teams).

• Architectural work + support to development, integration, V&V and product teams.
## Capability Engineering Dashboard/Kanban

### CE Dashboard

<table>
<thead>
<tr>
<th>Key Requirements in Progress</th>
<th>CoS</th>
<th>Value</th>
<th>Work Items Completed</th>
<th>% Work Items Completed</th>
<th>% Value Completed</th>
<th>Number of work items blocked</th>
<th>Expected Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Special Eng.</td>
<td></td>
<td>Average WIP</td>
</tr>
<tr>
<td>Requirement 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specialty 1</td>
<td></td>
<td>Additional Information</td>
</tr>
<tr>
<td>Requirement 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specialty 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specialty 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specialty 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specialty 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specialty 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specialty 7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Backlog (Demand) - Capability Analysis - Operational Concept Development - Capability Requirements Creation - Capability Development - Done

- **Backlog (Demand)**: Special Engineering Services
  - **In Progress**: Done, COTS (Rsrc 1 WIP, Rsrc 2 WIP), Security (Rsrc 3 WIP), Safety (Rsrc 4 WIP, Rsrc 5 WIP), Real-Time (Rsrc 6 WIP, Rsrc 7 WIP), Performance (Rsrc 8 WIP)
Product/Domain Engineering KSSs

• Separate KSSs for each product or domain team in the enterprise.

• Similar many software development organizations today, with the added requirement to perform systems engineering within the product or domain scope.

• Provide information to higher level KSSs and dashboards all the way to ESM level.
# Product Team Dashboard

<table>
<thead>
<tr>
<th>PT Dashboard</th>
<th>CoS</th>
<th>Value</th>
<th>% Features Completed</th>
<th>% Value Completed</th>
<th># Features blocked</th>
<th>Expected Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements in Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in Progress</td>
<td>CoS</td>
<td>Value</td>
<td>Reqt</td>
<td>Sourced to</td>
<td>Blocked?</td>
<td></td>
</tr>
<tr>
<td>Feature 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Flow among and between KSSs
New Capabilities

• Interface to a new health insurance company
  — requires capture of additional information about patients, diagnoses, and physician orders

• Integrate and analyze information from multiple patient telemetry systems to improve diagnostic capabilities
  — COTS option: Identify and evaluate any COTS data fusion products that apply to the telemetry devices, select the “best” one, then integrate it into the enterprise
  — If no COTS available for all telemetry systems, two options:
    o Change non-compatible telemetry systems for more compatible ones and use a COTS product to integrate/analyze the desired information
    o Develop a custom application to do the integration and analysis.
Upgrade and Enhancement

• User response improvement
  — system response time is unacceptably slow and is potentially putting patient safety at risk
  — evaluate alternatives for improving the user response time and recommend one or more for funding.

• Periodic upgrade of pharmacy formulary information
  — Data on formularies and drug interactions updated quarterly (subscription service)
  — Updates analyzed against existing DB structures, any necessary updates to the data structures made, data structure updates tested and deployed, then populated with updated data
Normal Capability Development

Insurance and Telemetry Issues Initiated

Stakeholders

ESM

Demand

Ext. Source Request

Formulary Update

Performance Issue Initiated

Users

Performance Issue Transferred

US

Demand

Ext. Source Request

Pharmacy DT

Demand

Ext. Source Request

Date Certain Work Item

Database DT

Demand

Ext. Source Request

Requirements to Teams

Normal CoS Work Items

Important CoS Work Item

SEPG-NA 2013

1 October 2013

39
Critical Issue: Interoperability Problem

• Feature to electronically send patient records to an external health care system was implemented, fully tested and seemed to function well during the first 30 days after deployment

• Late one night, a physician noticed that an important entry by external health care system not entered properly in the time log
• Provides
  — Aligned, unified view of work in progress and status of work
  — Predictability through measures easily SPC’d and projected
  — Value-based scheduling considers both system-wide and product priorities
  — Better use of C/SE resources; better servicing of product team SE needs

• Unlinks planning, scheduling, integration and deployment cadences

• Enhances decision making

• Supports continuous improvement

• Key Factor: Provides opportunity for right conversations, right people, right time
References


