Agenda

- The Case for Change
- Kanban Concepts
  - Core Practices
  - Foundational Principles
- Implementing Kanban
  - Visualization of Work
  - Flow of Work
  - Cadence
- Metrics and Reporting
- Improvements
  - Models
  - Policies
- Kanban Leadership
- Summary
Kanban Parking Lot

Not Started
- Objective
- Question
- Objective
- Question
- Comment

In Process
- Objective
- Comment
- Question

Done
- Comment
- Question
- Objective
- Question
Kanban Parking Lot

Not Started

In Process

Done
The Case for Change
What is Lean?

• The core idea is to maximize **customer value** while minimizing waste.
• A lean organization understands customer value and focuses its key processes to continuously increase it.
• The ultimate goal: provide value to the customer through a value creation process that has zero waste.
• Optimize the Whole

[Lean.org](lean.org)
Lean Concepts

- Relentlessly eliminate anything that isn’t adding value
- Eliminate time spent on what “we know” we’ll need in future
- Eliminate inefficient ways of working
- Optimize the whole system
- People doing the work know best how to do it
- Mapping processes and improving
- WoMBaT: Waste of Money, Brains, and Time

Kanban takes a lean thinking approach to improving processes.
What is Agile?

• Change Happens:
  – Priorities Changes
  – The Marketplace Changes
  – Requirements Change
  – Needs Change
  – People Change
  – Sponsorship Changes
  – Technology Changes

Your ability to react and respond to these changes is what really matters!

“I’m all for progress. It’s change I don’t like.”
Mark Twain
Why Agile?

Agile development uses an iterative approach to reduce risk and create products that are "launchable" throughout development.
Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

<table>
<thead>
<tr>
<th>Individuals and Interactions</th>
<th>over</th>
<th>Processes and Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Software</td>
<td>over</td>
<td>Comprehensive Documentation</td>
</tr>
<tr>
<td>Customer Collaboration</td>
<td>over</td>
<td>Contract Negotiation</td>
</tr>
<tr>
<td>Responding to Change</td>
<td>over</td>
<td>Following a Plan</td>
</tr>
</tbody>
</table>

That is, while there is value in the items on the right, we value the items on the left more.
Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity—the are of maximizing the amount of work not done is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agilemanifesto.org
Common Agile Methodologies

Is Kanban under the Agile umbrella?

Agile

Crystal
MSF4Scrum
Agile Unified Process

Kanban

Scrum
MSF4AS
DAD
SAFe
LeSS
What is Kanban?

kan·ban
ˈkänbän/
noun
a Japanese manufacturing system in which the supply of components is regulated through the use of a card displaying a sequence of specifications and instructions, sent along the production line.
– an instruction card used in a kanban system.
– plural noun: kanbans

Pronunciations
David Larabee
Kanban Background

- From family of “pull” systems
- Pull systems expose bottlenecks
- Creates slack in non-bottlenecks
- New work is “pulled” into system
- Lean thinking applied to software development
- Empirical Process
Kanban Call to Action

- Achieve sustainable pace of work and work-life balance
- Reduce stress for team members
- Improve software development process across teams
- Recognize team uniqueness
- Implement process change with minimum resistance
- Continuously improve
- Main reason to adopt is Change Management

“Prescriptively enforcing a software development process on a team didn’t work.”
- David J. Anderson, Author “Kanban, Successful Evolutionary Change for Your Technology Business”
Approach to Change

Old Performance

“We are waterfall”

“Kanban approach to change”

“A” Agile approach to change

New Performance

“We are agile”
Benefits

- Improve productivity
- Improve predictability
- Increase customer satisfaction
- Reduce delivery times
- Facilitates moving to a continuously improving organization
- Create more functional working relationships across organization
What Kanban Is Not --

• Not your traditional way to run projects
• Not a methodology but a way to continuously improve processes
• Not an approach to project management
• Not installation of an Agile method
• Not a lack of discipline
• Silver Bullet – Doesn’t fix everything

Kanban does not help you architect software or perform tests or write requirements.
When is Kanban a Good Fit?

• Uneven flow of work
  – Large batch transfers
  – Unplanned, speculative, disruptive requests
  – Blocking issues

• Deferred commitment is desirable
  – Priorities change frequently
  – Constant re-planning
  – High abandonment, discard, abort rates
  – Delivered work, never used

• System or workers are overburdened
  – Too much work-in-progress
  – Stressed workers
  – Poor quality
  – Long/unpredictable lead times, Long wait queues
When to Consider Other Options?

• Highly mature organization
  – Demand never exceeds capacity, flow is smooth and never interrupted, no overburden

• Facing extinction
  – No time to let Kanban work its magic, need revolution vs. evolution

• Boss lacks patience for incremental improvement to take effect
Kanban vs. Scrum

Kanban!!!

Scrum!!!
# Scrum vs. Kanban

<table>
<thead>
<tr>
<th>Scrum</th>
<th>Kanban</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Stories</td>
<td>Work Items (e.g. user stories)</td>
</tr>
<tr>
<td>Daily Standup: Focused on 3 questions</td>
<td>Daily Standup: Focused on flow of work</td>
</tr>
<tr>
<td>Scrum of Scrum occurs after Daily Scrum</td>
<td>Program Level meetings happen first</td>
</tr>
<tr>
<td>Fixed Iterations/Sprints</td>
<td>Continuous Flow</td>
</tr>
<tr>
<td>Feature delivery at end of Sprints</td>
<td>Feature delivery decoupled from Sprints</td>
</tr>
<tr>
<td>Velocity</td>
<td>Cycle Time</td>
</tr>
<tr>
<td>Story Point Estimations</td>
<td>Class of Service (Forget estimates!)</td>
</tr>
<tr>
<td>Encourages process conformity</td>
<td>Each team is unique</td>
</tr>
<tr>
<td>Roles: Product Owner, Scrum Master, Team</td>
<td>Use existing roles</td>
</tr>
<tr>
<td>Protect Sprint from change</td>
<td>Change can happen anytime</td>
</tr>
<tr>
<td>Delivery Mechanism</td>
<td>Change Mechanism</td>
</tr>
<tr>
<td>Burn Down Charts</td>
<td>Cumulative Flow Diagrams</td>
</tr>
<tr>
<td>Defined Process</td>
<td>Evolves Process</td>
</tr>
</tbody>
</table>
Lack of Roles is a Strength!

- No prescribed roles in Kanban!
- Roles remain same as today
- Build cross-functional skills
- Kanban Change Agent
  - Kanban Lead
  - Kanban Coach
  - Leads Kanban Initiative
  - Facilitates Kanban system design
  - Helps remove impediments
  - Servant Leader
Kanban & Scrum Teams

• **Useful Scrum concepts:**
  – **Scrum Roles**
    • Product Owner
    • Scrum Master (Kanban Change Lead)
    • Small Teams
  – **Scrum Meetings**
    • Daily Scrum
    • Retrospectives
    • Backlog Grooming
    • Demos
  – **Scrum Artifacts**
    • Product Backlog
Scrum Team Roles

**Development Team**
- Typically 6-9 people
- Cross Functional in order to build working software entirely by themselves
- Self-Organizing
- Keep work moving smoothly (everyone)

**Product Owner**
- Empowered by the organization to make decisions on behalf of the product
- Sole person responsible for managing the Product Backlog
- Product Owner is one person, not a committee

**Scrum Master**
- Servant Leader to the Development Team by removing impediments
- Ensure Kanban methods ceremonies are conducted
- Coach to the Development Team and Product Owner
Kanban Mindset

- Continuous Improvement
- Process Evolution
- Making the team successful
- Empowered team
- Openness and Visibility
- Collaboration
Kanban Concepts

- Visualize
- Work Items
- Card Walls
- WIP
- Change
- Workflow
- Cadences
- Lead Time
- Change
Kanban Core Practices

1. Visualize
2. Limit Work-in-Progress
3. Manage Flow
4. Make Policies Explicit
5. Implement Feedback Loops
6. Improve Collaboratively, Evolve Experimentally
Emergent Behaviors

- Process tailored to each project or value stream
- Decoupled Cadences
- Work scheduled by opportunity
- Value optimized with Class of Service
- Risk managed with Capacity Allocation
- Tolerance for Process Experiments
- Quantitative Management
- Viral spread of Kanban in organizations
- Small teams merged for liquid labor pools
1. Visualize

- Visualize Workflow
  - Make the invisible, visible
  - Mechanism
  - Interactions
  - Handoffs
  - Queues & Buffers

- Cards Walls
  - View of system
  - Visually track WIP
  - Self-organize, live collaboration
  - Near real-time project status
2. Limit Work-in-Progress

Setting Explicit Policies That Limit Work in Progress Kickstart a Virtuous Cycle of Improvement

- Reduce WIP Limits
- Reduce Task Switching
- Reduce Cycle Times
- Increase Quality
- Increase Feedback
- Increase Team Maturity

Starting with limiting WIP makes the need for other practices obvious

Shirley Ronen-Harel: Why limit wip – Working better within constraints
Team Exercise

Name Game

• How long does it take to write a name?
  – 1 name?
  – 5 names?

<table>
<thead>
<tr>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 name</td>
</tr>
<tr>
<td>5 names</td>
</tr>
</tbody>
</table>

• What factors influence that time?
Team Exercise

Name Game

Exercise Overview:
• Divide into teams of 4-6 people
• Roles: 1 Developer, rest are Customers
• Customers:
  – Customers can’t write
  – Want your name written as quickly as possible
  – Record time needed to get their name written
• Developers:
  – Developers have the skill to write
  – Must follow corporate policy
• Put away name tags!
Team Exercise

Round 1

- Only the Developer can write
- Must write all names at the same time, one letter at a time!
- Names must be correct or return
- Record Start Time
- When name finished, Customer records finish time
- At end, team chooses the Median time
- Record results

Company Policies:

1. Never keep a customer waiting
2. Start Early = Finish Early

When the timer starts, Customers hand their cards to the Developer at the same time and starts giving their names
Round 1 – Recap

- Metrics by Team
  - Median Time
  - Total Time

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Round 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 names</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- What influenced the time the most?
Team Exercise

Round 2

Company Policies:
1. Limit WIP (work-in-progress)
2. WIP Limit = 1 customer at a time

- Rotate Developers
- Developers can only work on 1 Customer at a time!
- Customer records Start Time and Finish Time for their name (calculate delivery time)
- At end, team chooses the Median time
- Record results

When the timer starts, the Developer will hold out their hand when they are ready for the next card for the name.
Team Exercise

Round 2 – Recap

• Metrics by Team
  • Median Time
  • Total Time

What are the results?

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Round 1</th>
<th>Round 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 names</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Company Policies:

1. Limit WIP (work-in-progress)
2. WIP Limit = 1 customer at a time

Next

Henrik Kniberg
Multi-Tasking BUSTED!
3. Manage Flow

• Start with Existing Processes

• Seek smoothness, timeliness, good economic outcomes

• Drives Improvement

• Focus on Flow vs. Waste

• Measure Work

• Work-in-Progress (WIP)
Chart Examples

Control Chart

Cumulative Flow
4. Make Policies Explicit

- Process Policies
- Explicit and Visible
- Build trust in the system
- Helps everyone understand what’s expected
- Enables team members to make decisions
- Let’s team decide how to keep work flowing
- Changes as teams evolves
Policy Examples

- Actions to be taken when finished coding
- Impediment/Blocker actions to be taken
- Production Bugs priority over QA Bugs, both priority over new development
- Work requested through different channel
- Who adds work to boards
- Ideas for design changes
- WIP Limits
- Definition of Done
- < 5 days dev
5. Implement Feedback Loops

• **Purpose:**
  – Compare expected outcomes to actual outcomes
  – Make process and policy adjustments

• **Feedback Loops:**
  – Standup Meeting
  – Service Delivery Review
  – Operations Review
  – Risk Review
6. Improve Collaboratively, Evolve Experimentally (using models and scientific method)

- Quantitative
- Scientific Approach to Improvements
- 3 Models:
  - Theory of Constraints
  - System of Profound Knowledge
  - Lean Economic Model
Implementing Kanban

- To Do: Use Kanban, Try Kanban Tool
- Doing: Learn about Kanban
- Done: Get some sticky notes, Get a whiteboard
Cultural Change vs Managed Change

- Not typical planned-transition implementation
  - Not big-planning-up-front-style
  - No planned initiative, no assessments
  - No declaration that “Now we’re Kanban!”
  - Ideally: there is no end
Kanban Values

- Respect
- Courage
- Focus on Value
- Communication & Collaboration
- Holistic Approach to Change
High Performing Teams

High Performance Tree
Getting Started

1. Agree on Goals
2. Process
   – Map the Process
   – Define Input Point
   – Define Exit Point
3. Work Item Types (WITs)
   – Define WITs
   – Analyze Demand for WITs
4. Card Wall
   – Create Card Wall/Board
   – Create Electronic Board (optional)
5. Feedback
   – Agree on Standup
   – Agree on Operations Review
6. Educate Team
1. Agree on Goals

**Kanban Goals**

**Business Goals**
- Improve Lead Time
- Optimize Existing Processes
  - Improve Time to Market
  - Control Costs

**Management Goals**
- Transparency
- Enable High Maturity
- Deliver High Quality
- Simplify Prioritization

**Organizational Goals**
- Improve Employee Satisfaction
- Provide Slack to Enable Improvement

**Examples:**
- Translucency
- Enable High Maturity
- Deliver High Quality
- Simplify Prioritization
Example: Goals

Planning
- Enterprise level planning and strategic priorities
- Forward looking project delivery plan
- Visible project status and performance metrics
- Project inter-dependencies identified
- Single point of entry for project requests
- Clear enterprise priorities

Project Management
- Consistency in project management
- Defined standards and checkpoints
- Techniques that drive innovation
- Balanced
- Proactive
- Cost control techniques

Happy teams
- Balanced work/life
- Foundational knowledge of methods
- Opportunities to grow as individuals

Highly motivated individuals and teams
- Supportive and involved executives

People & Culture
- Family oriented company
- Expectations aligned with performance management

Methodology
- Agile-based methods
- Lightweight and streamlined
- Consistently applied delivery models
- Standards based
- Full life cycle
- Options based on project type
- Understood by all

Quality
- Actionable performance metrics
- Zero Defects
- Highly supportable systems
- Thorough unit and systems testing
- Clear understanding of testing requirements
- Culture of continuous improvement

Predictability
- Expect change, manage change
- Delivery to commitments
- No surprises
- Predictable work week
- Increasingly accurate delivery dates
- Clear priorities, clear assignments, clear expectations
Team Exercise

Individually, write down:

• What are the goals you would like to realize with Kanban:
  1. For your team?
  2. For your organization?
  3. For yourself?

• Be prepared to share and explain your reasoning to the team
Kanban Bargain

• **Traditional Bargain:**
  – Promise to deliver based on scope-time-money
  – Estimation, planning, budget, requirements, schedule

• **Agile Bargain:**
  – Promise to deliver in iterations
  – Scope prioritized often, scope dropped if something has to give

• **Kanban Bargain:**
  – Delivery: Agree to regular delivery of high-quality software
  – Transparency: Process, daily visibility
  – Flexibility: Frequent opportunities to select most important items
  – Continuous Improvement: Team makes ongoing effort to increase delivery
  – Commitment: Against service level/cycle time
Kanban Foundations

Start with what you do now

Agree to pursue incremental, evolutionary change

Initially, respect current roles, responsibilities, and job titles

Encourage acts of leadership at all levels
2. Process

- Define the typical flow for the Workflow
  - Features, User Stories, Requirements, Work Packages, Services, Incidents, etc.
- Map the sequence from request to delivery
- Define Input Point
- Define Exit Point
- Build Card Wall
- Example:

  ASD 1.0  Envisioning  ASD 2.0  Planning  ASD 3.0  Developing  ASD 4.0  Stabilizing  ASD 5.0  Deploying  ASD 6.0  Improving

  Analyze  Develop  Accept
3. Work Item Types

- Define types of work that can enter process as input

- Agile Examples:
  - User Story
  - Bug
  - Quality of Service

- Waterfall Examples:
  - Requirement
  - Change Request

- Service Management Examples:
  - Incident
  - Problem
  - Service Request
4. Card Walls/Kanban Boards

• Visually depicts flow of work

• Tailor to reflect current process

• Changes with improvements to process

• Adopt how others use boards
Building a Board

How to Build a Kanban Board
<table>
<thead>
<tr>
<th>Analyze</th>
<th>Develop</th>
<th>Accept</th>
</tr>
</thead>
</table>

Initial Board Example

[Image of a table with stages: Analyze, Develop, Accept, and arrows pointing to each stage]
Group Exercise – Build Board

- Break into groups
- Each group creates an initial Kanban Board like the one here:

  - Use:
    - Whiteboard
    - Flip chart
    - Wall with tape
    - Table Top with tape

- Choose a Team Name
Group Exercise – Setup

• Pick Roles (arrange in order):
  – Product Owner
  – Analyst
  – Developer
  – QA
  – Customer (Instructor)

• Materials Needed:
  – Kanban Board
  – Sticky Notes (Product Owner)
  – Pens (everyone)
  – Catalog of Pictures (on slide)

Set the stage:
• 1 production “day” = 2 minutes
• Process in batches of 5
• Produce as many as possible
• 3 Simulated Production Days
1. Product Owner
   - Create cards for 5 pictures
     • Write Picture # on card
     • Put all 5 cards into “Analysis”
   - Start on next batch of cards

2. Analyst
   - Write “Title” of picture on all 5 cards
   - Put all 5 Cards into “Development”

3. Development
   - Draw picture on card for all 5 cards
   - Put all 5 Cards into “Accept”

4. QA
   - Inspect all 5 cards
   - Remove cards from board

5. Write down the number of Pictures produced that day
• Each Group discusses the 3 “Days”
  – How many Pictures were produced each day? (don’t count incomplete)
  – What was the Throughput?
  – What problems were encountered?
  – Report Out

<table>
<thead>
<tr>
<th>Day</th>
<th>Pictures Produced</th>
<th>Throughput Pictures/2 Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average &gt;&gt;&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Queues and Buffers

<table>
<thead>
<tr>
<th>Ready</th>
<th>Analyze</th>
<th>Develop</th>
<th>Accept</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Analyze</td>
<td>Dev</td>
<td>Ready for Accept</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ready for Dev</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Input Queue**

**Buffers**
Swim lanes can be created as needed to represent different Work Item Types, flows of work, teams, projects, product, feature, epic, etc.

Swim Lanes represent different streams of work.
## Record Entry/Exit Criteria

<table>
<thead>
<tr>
<th>Ready</th>
<th>Analyze</th>
<th>Develop</th>
<th>Accept</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Analyze</td>
<td>Dev</td>
<td>Ready for Accept</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ready for Dev</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bug</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Criteria
- Design Complete
- Test Case Examples Done
- UIX Input Ready
- Code Complete
- Source Checked In
- Unit Tests Green
- Build Successful
- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete
## Basic Board Example

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (5)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Analyze</td>
<td>Dev</td>
<td>Accept</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ready for Dev</td>
<td>Ready for Accept</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Bug</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Criteria
- Design Complete
- Test Case Examples Done
- UIX Input Ready

- Code Complete
- Source Checked In
- Unit Tests Green
- Build Successful

- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete
## Basic Board Components

### Workflow Steps

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (3)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
</tbody>
</table>

### Bug

### Criteria

- Design Complete
- Test Case Examples Done
- UIX Input Ready
- Code Complete
- Source Checked In
- Unit Tests Green
- Build Successful
- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete
Work-in-Progress

- Work-in-Progress includes:
  - Number of “To-Dos” in your day
  - Number of User Stories being developed
  - Amount of multi-tasking

- Initially a guess
  - Adjust to achieve maximum flow

- Adjust based on flow:
  - Work Backed Up = Lower WIP
  - Idle Time = Increase WIP

- WIP Limit may be:
  - Number items (e.g. user stories, service tickets, etc.)
  - Story Points
  - Hours
Work-in-Progress

- **Methods to Limit WIP**

<table>
<thead>
<tr>
<th>Organizing work by type</th>
<th>Backlog</th>
<th>Analysis &amp; Design</th>
<th>Code</th>
<th>Test</th>
<th>Deploy</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Doing</td>
<td>Ready</td>
<td>Doing</td>
<td>Ready</td>
<td>Doing</td>
<td>Doing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Organizing work to smallest level</th>
<th>Backlog</th>
<th>Analysis &amp; Design</th>
<th>Ready to Code</th>
<th>Code</th>
<th>Ready to Test</th>
<th>Test</th>
<th>Ready to Deploy</th>
<th>Deploy</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Doing</td>
<td>Ready</td>
<td>Doing</td>
<td>Ready</td>
<td>Doing</td>
<td>Ready</td>
<td>Doing</td>
<td>Doing</td>
<td>Done</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Organizing work to make decisions</th>
<th>Backlog</th>
<th>Analysis &amp; Design</th>
<th>Code</th>
<th>Test</th>
<th>Deploy</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Doing</td>
<td>Done</td>
<td>Doing</td>
<td>Done</td>
<td>Doing</td>
<td>Done</td>
</tr>
</tbody>
</table>

NetObjectives.com
# Basic Board Components

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (5)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
</tbody>
</table>

**Bug**

- Design Complete
- Test Case Examples Done
- UIX Input Ready

**Criteria**

- Code Complete
- Source Checked In
- Unit Tests Green
- Build Successful
- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete

**WIP Limits**
## Basic Board Components

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (5)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
</tbody>
</table>

- **Analyze**: Ready for Dev
- **Develop**: Ready for Accept
- **Accept**: Feature

### Criteria
- **Feature**
  - Design Complete
  - Test Case Examples Done
  - UIX Input Ready
- **Code Complete**
  - Source Checked In
  - Unit Tests Green
  - Build Successful
- **Acceptance Tests Green**
- **Manual Testing Okay**
- **PO Acceptance**
- **Doco Complete**

### Bug

### Split Work: Doing, Done
## Basic Board Components

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (5)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
</tbody>
</table>

### Criteria for “Done”
- Design Complete
- Test Case Examples Done
- UIX Input Ready
- Code Complete
- Source Checked In
- Unit Tests Green
- Build Successful
- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete
### Basic Board Demo

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (5)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
</tbody>
</table>

**Bug**

- Design Complete
- Test Case Examples Done
- UIX Input Ready
- Code Complete
- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete

**Criteria**

- All Queues are Full: No more work can be added

---
### Basic Board Demo

#### Ready (5)
- Feature
- Feature
- Feature
- Feature
- Feature

#### Analyze (3)
- Feature
- Feature
- Feature

#### Develop (5)
- Feature
- Feature
- Feature
- Feature
- Feature

#### Accept (3)
- Feature
- Feature
- Feature

#### Ready for Release
- Feature
- Feature
- Feature

### Criteria
- Design Complete
- Test Case Examples Done
- UIX Input Ready
- Code Complete
- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete

---

**Ready to fill and re-sequence**
Group Exercise – Update Board

• Break into groups
  – Change roles if desired

• Each group updates their Kanban Board:
  – Input Queue
  – Analyze Buffer
  – Develop Buffer
  – Done Column
  – WIP Limits

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze</th>
<th>Develop</th>
<th>Accept (3)</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Exercise K2**
1. **Product Owner**
   - Create card with Picture #
   - Put card into “Ready”
   - Start on next card

2. **Analyst**
   - Pull card from “Ready”
   - Write “Title” of picture on card
   - Put card into “Ready for Dev”

3. **Development**
   - Pull card from “Ready for Dev”
   - Draw picture on card
   - Put card into “Ready for Accept”

4. **QA**
   - Pull card from “Ready for Accept”
   - Inspect
   - Put card in “Done”

5. **Record number of Pictures produced**
<table>
<thead>
<tr>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
<th>Round 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Mail Box</td>
<td>23. Stop Sign</td>
<td>43. Square</td>
<td>63. Tall Building</td>
</tr>
<tr>
<td>4. Circle</td>
<td>24. Stick Woman</td>
<td>44. Football</td>
<td>64. Rain</td>
</tr>
<tr>
<td>10. Dog</td>
<td>30. Guitar</td>
<td>50. Key</td>
<td>70. Heart</td>
</tr>
<tr>
<td>12. Fish</td>
<td>32. Door</td>
<td>52. Graph</td>
<td>72. Keyboard</td>
</tr>
<tr>
<td>13. Hat</td>
<td>33. Box</td>
<td>53. Arrow</td>
<td>73. Funnel</td>
</tr>
<tr>
<td>14. Stick Man</td>
<td>34. Stick Dog</td>
<td>54. Stick Cat</td>
<td>74. 4 Leaf Clover</td>
</tr>
<tr>
<td>15. Hand</td>
<td>35. Road</td>
<td>55. Bicycle</td>
<td>75. Eyes</td>
</tr>
</tbody>
</table>
Group Exercise – Round 2 Review

• Each Group discusses the 3 “Days”
  – How many Pictures were produced each day? (don’t count incomplete)
  – What was the Throughput?
  – What results were achieved?
  – Report Out

<table>
<thead>
<tr>
<th>Day</th>
<th>Pictures Produced</th>
<th>Throughput Pictures/2 Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>&gt;&gt;&gt;&gt;&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Kanban Board Examples
Example Work Item Card

User Story: As a persona I want something for some reason.

Queue Entry: 
Start Date: 
Finish Date: 
Due Date: 

Estimate: S/M/L

Tracking:
### Tracking Work Item Examples

<table>
<thead>
<tr>
<th>Tracking #:</th>
<th>Work Item Type:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title:</th>
<th>User Story: As a persona I want something for some reason.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UI Design, Test Cases, Coded, Unit Test</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue Entry:</th>
<th>Estimate: S/M/L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Date:</td>
<td></td>
</tr>
<tr>
<td>Finish Date:</td>
<td></td>
</tr>
<tr>
<td>Due Date:</td>
<td></td>
</tr>
</tbody>
</table>

#### Tracking:
- **Blocked Item**
- **Assignee**
- **SLA Warning**
**Blocked Item**

- **Blocked Item:**
  - An item that is clogging the pipeline of work through the system
  - “Blocker”
  - Special cause variation

- **Different from a Bottleneck:**
  - Bottleneck is a process flow restriction
  - Blocked item is a specific item impeding flow

- **Handling Blockers:**
  - Organization must have capability to restore flow
  - Root cause analysis
  - Track as “Issue” work item, generally attach Red or Pink ticket to card
Bottlenecks

- Process flow where a backlog of work builds up pending processing
- Impedes workflow
Issue Management and Escalation Policies

• Issue Work Item:
  – Team member marks item as Blocked
  – Records Issue
  – Attaches Pink ticket with info
  – Discussed at Daily Standup
    • Determine Assignee: e.g. Project Manager, Idle Team Member

• Escalation:
  – Team unable to resolve Issue
  – Escalation Policy needed
  – Report issues over time
Swarming

- More people focused on a situation
- Dog pile the problem
- Examples:
  - Blocker (team)
  - Meet a release date
  - Urgent production problem
Electronic Tracking
5. Feedback

7 Kanban Cadences

David J. Anderson, Kanban
Cadences

- **Common Meetings:**
  - Standup Meeting
  - Replenishment Meeting
  - Delivery Planning Meeting
  - Operations Review

- **Recommended:**
  - Product Demo
    - Schedule, definitely before a release
  - Retrospective
    - Schedule, e.g. bi-weekly or monthly reviews

- **Other Possible Meetings:**
  - Strategy Review
    - Product Strategy, current markets
  - Risk Review
    - Review Blockers and Review Lead Time outliers
  - Service Delivery Review
    - Focused on system capability

✅ Successful Practice: No more than 10% of time spent in meetings!
Standup Meetings

• Purpose: Team reviews work-in-progress and coordinates work for the day
  – Focus on flow of work

• Cadence: Daily

• Length: 15 minutes

• Attendees:
  – Team
  – Invite stakeholders but don’t mandate attendance
Different Standup Formats

• Informal, no agenda or structure (not desirable)

• Manager Interrogation (e.g. Project Manager asks for updates)

✓ Around Room (e.g. Scrum format)
  – What I did yesterday
  – What I will do today
  – What impediments I have

✓ Kanban Board: Review Work Items
  – Right to left, start with items close to completion, end meeting when no use looking further upstream

✓ Kanban Board: Review Blockers and at risk items
  – Right to left, discuss just blocked items or items that are at risk

⚠ Successful Practice: Leverage the boards!
Standup Meeting Example

• Before Standup
  – Team members update their active items
  – Leader updates Cumulative Flow

• During Standup
  – Do we have a bottleneck?
  – Do we have a blocker?
  – Are we keeping WIP limits?
  – Are priorities clear?
  – What did we do yesterday?
  – What are we planning today?

• After Standup?
  – Update charts
  – Huddles on specific items, features, issues
  – May break into a spontaneous Scrum
Queue Replenishment Meeting

• Purpose: Re-fill input queue with new, prioritized items
  – Prioritization deferred to the last reasonable moment, when it’s put on the board
  – Feedback from Customer on needs

• Cadence: Frequently, Weekly

• Attendees:
  – Business Owners/Product Owner (with items in backlog)
  – Delivery Manager (e.g. Project Manager)
  – Potential Stakeholders:
    • Development/Test/Technical Manager
    • Architect (assess technical risk)
    • Usability
    • Business Analyst
    • Operations

• Replenishment meetings have many different formats and are context dependent
  – Internal customers, external customers, proxy customers
  – Product Owner

⚠️ Successful Practice: Hold replenishment meetings frequently!
The whole team is responsible for progressing work items.
Delivery Planning Meeting

• Purpose: Plan downstream delivery
  – Release Planning
  – Product Owner presents release goals

• Cadence: Based on delivery cycle
  – E.g. Releases every 2 weeks, Fixed date releases

• Attendees:
  – Delivery Manager (e.g. Project Manager)
  – Business Owners/Product Owner
  – Potential Stakeholders:
    • Development/Test/Technical Manager
    • Team
    • Operations

• Input from Strategy Review
  – E.g. Product Strategy, Lean Startup
Example – Fixed Number of Features

If “scope” or features are fixed, map out a release date based on cycle time.

- Items in Release: 100 items
- WIP Limit: 16 items
- Cycle Time: 14 days

Duration until Release = 87.5 days

- **Cycle Time:**
  - Time Actually Spent on the Item
  - Started Until Done
- **Work-in-Progress:**
  - Number of items active (in progress) at a point in time
Example – Fixed Delivery Date

Release date fixed, how many features will be included in based on cycle time.

16 items
WIP Limit

×
Cycle Time
14 days

= Duration until Release
28 Days

= Items in Release
32
On Demand and Ad Hoc Deliveries

• Examples of circumstances warranting ad hoc deliveries:
  – Low-cost coordination costs of delivery (e.g. mature organization)
  – New continuous deployment (e.g. startups)
  – Urgent request (e.g. critical production defect)
  – Off-cycle release (e.g. customized software for major customer)
Operations Review

• **Purpose:**
  – Disciplined review of demand and capability of each Kanban system
  – Keystone to organizational transition
  – Foster Kaizen
  – System of Systems
  – Suggest improvements

• **Cadence:** Monthly

• **Attendees:**
  – Multiple Kanban teams
  – Management

• **Specific Topic Examples:**
  – Guests (add interest)
  – Department Update (e.g. 8 min. each)
  – Team Updates with metrics (e.g. 5 min. each)

Team Updates:
- Defect Rates
- Lead Time
- Throughput
- Issues Review
- Value-added Efficiencies
- Special Reports
Demo

- Review of work completed by the team with Business Owner, Customer, Product Owner
  - Primarily for Stakeholders to solicit feedback
- All work completed should be reviewed with an emphasis on quality and completeness
- Feedback from the review comes in the form of new tasks and re-prioritization
- Before Releases at a minimum
Retrospective

• **Purpose:**
  – The Retrospective is an opportunity for the team to inspect itself and create a plan for improvements to be enacted.

• **Cadence:**
  – Schedule regularly, every 2 weeks, no more than monthly
  – As needed

• **Agenda:**
  – What went well
  – What could have gone better
  – 3-5 things to improve

• **Ad Hoc Retrospectives**
  – Conducted as needed to address issues or at key milestones of long projects

*The objective is to LEARN from the experience by facilitating a very open, blame-free discussion of successes and mistakes.*
Group Exercise – Update Board

Exercise K3

• Break into groups
  – Change roles if desired
• Discuss the previous two rounds, state of current board (Daily Standup + Retrospective)
• Make any changes to board (e.g. WIP, capacity, Swarming)
Group Exercise – Round 3

1. Product Owner
   - Create card with Picture #

2. Analyst
   - Pull card, add “Title”

3. Development
   - Pull card, Draw picture

4. QA
   - Pull card, Inspect

5. At end of each day:
   - Record Pictures produced
   - Perform Daily Standup/Retrospective
   - Tune system

Set the stage:
- 1 production “day” = 2 minutes
- Process 1 card at a time
- 3 Production Days
- Produce as many as possible
- There may be curve balls from the Customer!
<table>
<thead>
<tr>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
<th>Round 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Mail Box</td>
<td>23. Stop Sign</td>
<td>43. Square</td>
<td>63. Tall Building</td>
</tr>
<tr>
<td>4. Circle</td>
<td>24. Stick Woman</td>
<td>44. Football</td>
<td>64. Rain</td>
</tr>
<tr>
<td>10. Dog</td>
<td>30. Guitar</td>
<td>50. Key</td>
<td>70. Heart</td>
</tr>
<tr>
<td>12. Fish</td>
<td>32. Door</td>
<td>52. Graph</td>
<td>72. Keyboard</td>
</tr>
<tr>
<td>13. Hat</td>
<td>33. Box</td>
<td>53. Arrow</td>
<td>73. Funnel</td>
</tr>
<tr>
<td>14. Stick Man</td>
<td>34. Stick Dog</td>
<td>54. Stick Cat</td>
<td>74. 4 Leaf Clover</td>
</tr>
<tr>
<td>15. Hand</td>
<td>35. Road</td>
<td>55. Bicycle</td>
<td>75. Eyes</td>
</tr>
</tbody>
</table>
• Each Group discusses the 3 “Days”
  – How many Pictures were produced each day? (don’t count incomplete)
  – What was the Throughput?
  – What could be improved?
  – Report Out

<table>
<thead>
<tr>
<th>Day</th>
<th>Pictures Produced</th>
<th>Throughput Pictures/2 Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>&gt;&gt;&gt;</td>
</tr>
</tbody>
</table>
6. Educate Team

• Prepare and Reassure Team

• Topics:
  – Kanban Concepts
  – Card Board, Cards
  – Managing WIP
  – WIP Limits
  – Class of Service
  – Pull Work

• Level Set:
  – Nothing else changes
  – Same Job Descriptions
  – Same Activities
  – Same Handoffs
  – Same Artifacts
  – Same Process
Kanban Metrics and Reporting
Kanban Metrics

• Different from other methodologies, including Agile
  – Kanban changes way team interacts

• Focused on flow of work
  – Less focused on project “on-time” or plan being followed

• Goals:
  – Smooth Work Flow
  – Predictability
  – Operating as designed
  – Continuous improvement
Common Kanban Metrics

- Work-in-Progress
- Lead Time
- Cycle Time
- Due Date Performance
- Throughput
- Issues Tracking
- Flow Efficiency
- Initial Quality
Cumulative Flow Diagram

- **Information Radiator**
  - Up-to-date status of flow of work

- **Visual display of development progress over time**

- **Reflects “states” defined on Kanban Board**

- **Tracks completion progress by specified UOM (# items, story points, hours, etc.)**

- **Average Lead Time, Average Cycle Time, Work-in-Progress**
Flowing WIP

Cumulative Flow Diagram

Smoothly flowing Work-in-Progress
Bands of work remain consistent
Bottlenecks

Dev waiting on Analyze
Dev not achieving WIP limit of 5
Load Balancing Strategies

- Add more people
- Off-load people who are constraints
- Help people who are constraints
- Have others help out on constraints
- Improve the workflow
- Create teams
Cycle Time and Lead Time

• Lead Time:
  – Time Item Requested Until Done

• Cycle Time:
  – Time Actually Spent on the Item
  – Started Until Done

• Work-in-Progress:
  – Number of items active (in progress) at a point in time

• Cycle Time and Release Planning

100 items (Items in Release) / 16 items (WIP Limit) \( \times \) 14 days (Cycle Time) = 87.5 days (Duration until Release)
Lead Time

• How predictable do we deliver?

• Spectral analysis provides a broader range of information
Due Date Performance

- Fixed Delivery work items
  - Services, regulatory, etc.

- 13 months data for comparison

- Were items delivered on time?

![Due Date Performance Chart](image)
Throughput

- Throughput:
  - Trend of output from a process in a given period of time

- Cycle Time:
  - Length of time to complete a process
  - Becomes SLA with business

- Throughput = WIP/Cycle Time
Issues Tracking

Issues Cumulative Flow Diagram

- Blocked Item
- Active Issue
- Resolved Issue
- Closed Issue
Other Metrics

• Flow Efficiency
  – Comparison of Touch time to Wait time

• Initial Quality
  – Tracking of defects from completed features

• Failure Load/Cost of Poor Quality
  – Percent of work processed as a result of earlier poor quality
Scaling Kanban
Personal Kanban

- Kanban applied to one’s personal workload
- Choose the right work at the right time
  - Visualize your work
  - Limit your work-in-progress

Mike Burrows, Kanban from the Inside
Portfolio Kanban

• Kanban applied to Project Portfolios

• Think creatively about organizational problems

• Getting Started:
  – Start with what you do now
  – Find ways to:
    • Organize portfolio visually (program, team, customer…)
    • Limit WIP
    • Manage for smoothness and timeliness
    • Evolve decision framework
    • Collaborate

⚠️ Successful Practice: Stop starting and start finishing!
Scrumban

- Kanban combined with Scrum
  - Inside Scrum to drive team improvements
  - Outside Scrum to address challenges of scale

- Start with what you do now and leverage Kanban

- Transformation:
  - Visual process, build board
  - Standups center around board
  - Sprint WIP limits to continuous flow
  - Releases de-coupled from Sprint Planning
  - Sprint Planning gets easier

"Kanban with Scrum" instead of "Kanban vs. Scrum"

Mike Burrows, Kanban from the Inside
Large Projects

• Kanban on Large Projects
  – Lots of requirements
  – Large team sizes
  – Long periods of time between releases

• Start with what you do now

• Tips:
  – Hierarchical Requirements: Only track top two levels on board (not Tasks)
  – Identify Release Goal on right side of board so it’s visible and provides focus
  – May need additional Work Item Types
  – To manage flow, break requirements down to small similar sizes, like user stories or functional specs (.5 to 4 days)
    • Track large requirements with one color, breakdown requirements in another color
    • Limit WIP at both large and small requirement level
Minimal Marketable Release

- **Minimum Marketable Feature (MMF)**
  - Specific feature released

- **Minimal Marketable Release (MMR)**
  - Much larger than MMFs
  - May be collection of MMFs
  - Helps focus at a broader Release level
  - Leverage transaction costs
  - First MMR usually large

Theleanagilepmo.com
Two-Tiered Card Wall Example

<table>
<thead>
<tr>
<th>MMF (3)</th>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (5)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Analyze</td>
<td>Ready for</td>
<td>Dev</td>
<td>Ready for Accept</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dev</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMF 1</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
</tr>
<tr>
<td></td>
<td>Feature</td>
<td></td>
<td>Feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMF 2</td>
<td>Feature</td>
<td>Feature</td>
<td>Feature</td>
<td></td>
<td>Feature</td>
</tr>
<tr>
<td></td>
<td>Feature</td>
<td></td>
<td>Feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMF 3</td>
<td>Feature</td>
<td></td>
<td>Feature</td>
<td></td>
<td>Feature</td>
</tr>
<tr>
<td></td>
<td>Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Criteria**
- Design Complete
- Test Case Examples Done
- UIX Input Ready

- Code Complete
- Source Checked In
- Unit Tests Green
- Build Successful

- Acceptance Tests Green
- Manual Testing Okay
- PO Acceptance
- Doco Complete

**Release Goal:**
New Web Interface
Systems Integration

• Cross-team dependencies
• Project schedule depends on delivery from another team (or vice versa)
• Both teams have a card
• Treat as “Fixed Date Delivery” work item types
• Dependent Team: Treat as Blocker as it nears due date
Shared Resources

• Resources shared from another pool (e.g. architecture, security, specialist)

• Three methods to handle:
  – Simple Visualization: ID resource with smaller ticket/dot with name of resource, monitor quantity
  – Treat as a Blocked item
  – **Shared resource team creates own Kanban system** (e.g. Security, Data Architecture, etc.)

• Service-Oriented Architecture
  – Emerges as teams manage services and coordinate work using Kanban
Enterprise Kanban

Strategy
- Fitness for Purpose
- Market Segmentation
- Capability Analysis
- Strategic Alignment
- Portfolio Management
- Risk Hedging
- Class of Service Definition
- Fitness Criteria Metrics

Risk Management
- (Custom) Risk Profiles
- Business Risk Assessment
- Service Dependencies
- Technical Risk Assessment
- Blocker Clustering
- Project Risk Management

Planning
- Capacity Planning
- Capacity Allocation
- Forecasting
- Simulation
- Selection & Commitment
- Sequencing

Resilience/Survivability
- Sense & Respond
- Evolutionary Change

Scheduling
- Optimal Start
- Window of Opportunity
- Cost of Delay
- Booking Capacity

Demand Management
- Demand Analysis
- Demand Shaping
- Replenishment Policies
- Options

Service Delivery
- Kanban Systems
- Flow Management
- Labor Pool Liquidity
- Service-oriented Organization
- Upstream Kanban
- Capacity Allocation
- Workflow Liquidity
- Shared Services
- Portfolio Kanban
- Service Delivery Review
- Operations Review
- Dependency Management

David J. Anderson & Associates, djaa.com
Group Exercise – Update Board

• Break into groups
  – Change roles if desired
• Discuss the previous rounds, state of current board (Daily Standup + Retrospective)
• Make any changes to board or strategy (e.g. WIP, capacity, Swarming)
Group Exercise – Round 4

1. Product Owner
   – Create card with Picture #

2. Team Analyst
   – Pull card, add “Title”

3. Development
   – Pull card, Draw picture

4. QA
   – Pull card, Inspect

5. At end of each day:
   – Record Pictures produced
   – Perform Daily Standup/Retrospective
   – Tune system

Set the stage:
- 1 production “day” = 2 minutes
- Process 1 card at a time
- 3 Production Days
- Respect capacity limits
- Help others with bottlenecks
- Produce as many as possible
Group Exercise – Round 4

Exercise K4

The Twist

- Each person uses a coin to control actions
- Toss the coin before making a move
- To indicate a blocked item, write a “B” on the card, cross out the “B” to unblock

<table>
<thead>
<tr>
<th>Heads</th>
<th>Tails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose one of the following actions:</td>
<td>Do both of these items (if possible):</td>
</tr>
<tr>
<td>- Advance one of your unblocked items</td>
<td>- Block one of your unblocked items</td>
</tr>
<tr>
<td>- Unblock one of your items</td>
<td>- Start new work item from buffer</td>
</tr>
<tr>
<td>- Start a new work item from buffer</td>
<td></td>
</tr>
<tr>
<td>- If you have no other options, pair up and help someone</td>
<td></td>
</tr>
</tbody>
</table>
## Picture Catalog

<table>
<thead>
<tr>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
<th>Round 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Mail Box</td>
<td>23. Stop Sign</td>
<td>43. Square</td>
<td>63. Tall Building</td>
</tr>
<tr>
<td>4. Circle</td>
<td>24. Stick Woman</td>
<td>44. Football</td>
<td>64. Rain</td>
</tr>
<tr>
<td>7. Tree</td>
<td>27. Bottle</td>
<td>48. Sun</td>
<td>68. Umbrella</td>
</tr>
<tr>
<td>11. Telephone</td>
<td>31. Cloud</td>
<td>52. Graph</td>
<td>72. Keyboard</td>
</tr>
<tr>
<td>12. Fish</td>
<td>32. Door</td>
<td>53. Arrow</td>
<td>73. Funnel</td>
</tr>
<tr>
<td>13. Hat</td>
<td>33. Box</td>
<td>54. Stick Cat</td>
<td>74. 4 Leaf Clover</td>
</tr>
<tr>
<td>14. Stick Man</td>
<td>34. Stick Dog</td>
<td>55. Bicycle</td>
<td>75. Eyes</td>
</tr>
<tr>
<td>15. Hand</td>
<td>35. Road</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Heads

Choose one of the following actions:
- Advance one of your unblocked items
- Unblock one of your items
- Start a new work item from buffer
- If you have no other options, pair up and help someone

### Tails

Do both of these items (if possible):
- Block one of your unblocked items
- Start new work item from buffer
• Each Group discusses the 3 “Days”
  – How many Pictures were produced each day? (don’t count incomplete)
  – What was the Throughput?
  – What could be improved?
  – Report Out

<table>
<thead>
<tr>
<th>Day</th>
<th>Pictures Produced</th>
<th>Throughput Pictures/2 Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>&gt;&gt;&gt;&gt;&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Making Improvements

Kanban Method in a Nutshell

Start where you are  
Shared and Visual Understanding  
Evolutionary Improve
Improvement Opportunities

• 3 common models to drive improvement
  – Theory of Constraints
  – Lean Economic Model
  – System of Profound Knowledge
Theory of Constraints

- Management paradigm: A very small number of constraints limits a manageable system from achieving its goals.

- POOGI (Process Of OnGoing Improvement)

- Thinking Process (TP)
  - Change and Change Resistance focus

FIVE FOCUSING STEPS

Step 0 sometimes added: Define the system’s goal.
Lean Economic Model

• Value, Value Stream, Flow
• Eliminating waste
  – Muda: Waste in any form, caused by –
    • Mura: Waste from unevenness, overburden, strain
    • Muri: Demand that exceeds process capacity
• 5 Whys (Beware! 5 Blames)

5 Improvement Steps

1. Identify Value
2. Identify Value Stream
3. Create Flow
4. Establish Pull
5. Identify Waste

“The most dangerous kind of waste is the waste we do not recognize.”
Shigeo Shingo
System of Profound Knowledge

- Deming’s Statistical Process Control developed into a management technique

- Variability caused by Common Cause and Special Cause

- Six Sigma derived from Deming
  - Data driven approach and methodology for eliminating defects
  - Drives towards six standard deviation threshold
Fishbone Diagrams
6 Sigma Black Belt Career Day
Estimations

• Different approach to forecasting
• Estimation considered a waste
• Controversial Topic
• Decide for your own implementation
  – Start with what you do now
• Does not mean estimates are not done
  – Do for the right reasons, useful, meaningful

#NoEstimates !!!
The Estimation Problem

• Estimates are still guesses
  – Unknown Requirements \times Unknown Effort = SWAG

• Estimations are engrained in the history of our work

• On-Time/On-Budget should be viewed as worst case scenarios

• Factors: Informed/Uninformed Opinions, Sandbagging, Compromises

• Estimations are wrong
• Estimate can be useless
• Estimates can be wasteful
• Estimates can be harmful

Why do we kid ourselves that it makes sense?
#NoEstimates

No effort estimates

Effortless estimates

No estimates of effort
Monte Carlo Simulation

• Technique used to approximate outcomes using simulation
  – Better to model projects as a flow of work items thru system

• Forecast anything
  – Weather, sales, commissions, projects

• Tool to:
  – Improve forecasting
  – Identify Priorities
  – Create more reliable forecasts
  – Increase confidence in forecasts
Z Curve Forecasting

- Gather historical data for similar effort
- Assess requirements for risk categories (classes of service)
- Apply distribution curves
- Use historical Data for similar effort

![Z Curve Diagram](image-url)

- Setup
- Productivity
- Cleaning Up
Example

Takt Time by legs of Z Curve
- Average time between deliveries
- Probably distribution curve applied
- Not a single number, distribution

Monte Carlo Simulation
- Work items by Z Curve leg
- Gives time to deliver project

Mode = 76 days; Median = 77 days; Mean = 78 days; 85th perc = 90 days
Estimation Summary

• #NoEstimates gets a lot of discussion!
• Team ultimately decides
• Valuable part of estimating is the conversation
• Methods exist to help with forecasting
• Other methods:
  – Small Enough, Too big
  – Big, Small
  – Stack by relative size, class of service
Service Level Agreements

- **Class of Service (CoS):** Defines different types of work

- Work classified to optimize Customer Satisfaction, economically

- CoS reduces need for detailed estimate or analysis

- Clearly display, e.g. card colors, swim lanes

- Define Policies by each CoS

- Allocate capacity to each CoS based on demand

- Train team members

- Enables self-organization, empowers team
Class of Service Examples

- **Expedite:**
  - Urgent work items, drop all else
  - e.g. Production Defect

- **Fixed Delivery Date:**
  - Work items required by specific date, usually penalty
  - e.g. Regulatory

- **Standard:**
  - Delivered according to policy

- **Intangible:**
  - Capability improvements, market experiments, usually medium to long-term

- **Other Examples:**
  - Innovation
  - Maintenance
  - Support
Class of Service

- Assignment of CoS
  - CoS assigned when selected for input queue
  - Based on prioritization method (e.g. Backlog)

- Define for each Kanban system

- Everyone should understand CoS

- Generally only 4-6 classes, keep small and simple

- Precise definition, unambiguous
Policy Examples

• Expedite:
  – Use white cards (or other, specify)
  – Only 1 Expedite request at any time
  – Qualified resource pulls Expedite requests immediately, all other work goes on hold

• Fixed Delivery Date
  – Use purple cards
  – Delivery date at bottom right-hand of card
  – Fixed date items receive some analysis and estimation

• Standard Class of Service
  – Use yellow cards
  – FIFO: Pull the oldest standard class item from the queue first
  – Standard class items are generally delivered x days

Guideline: No more than 6 policies per CoS.
Kanban Change

Toyota Production System

High Quality, Low Cost, Short Lead Time

Just-In-Time
- Pull System
- Takt Time
- Continuous Flow

Jidoka
- Stop and notify of abnormalities
- Separate man’s work and machine’s work

Heijunka
Standardized Work
Kaizen

Stability (4Ms)
(4Ms: Man, Material, Method, Machine)

Jamie Bonini, Toyota
Cultural Change

• Cultural change may be biggest benefit of Kanban

• Highest CMMI level of organizational maturity is “Optimizing”

• Kaizen Culture

“I’m all for progress. It’s change I don’t like.”
Mark Twain
Kaizen Culture

• Continuous improvement culture
• Workforce empowered and self-organized
• Tolerance of failure in name of process/performance improvement
• Collaboration and performance of team
• Visual Controls
• Volunteers for tasks
• Trusting Culture

Kaizen
改善
To make better
Kaizen Mindset

Everything can and should be improved

Not a single day should go by without some kind of improvement being made somewhere in the company

Imagine the ideal customer experience and strive to provide it

Don’t criticize, suggest an improvement

Think of how to improve it instead of why it can’t be improved

Think beyond common sense; even if something is working, try to find the ways to make it work even better

See problem solving as cross-functional systemic and collaborative approach
Agile Leadership is the ability of a leader to be able to lead well in a wide range of circumstances especially new, changing and ambiguous situations.

Agile leaders, or those with a high degree of Learning Agility, share some key characteristics including:
- Self-Awareness
- Mental Agility
- People Agility
- Change Agility
- Results Agility
Leadership Attributes

- Ambiguity Tolerance
- Curiosity
- Creativity
- Courage
- Conviction
- Emotional Resilience
- Critical Thinking
- Vision
- Flexibility
Servant Leadership

• Servant-leaders achieve results for their organizations by giving priority attention to the needs of their colleagues and those they serve.
• Servant-leaders are often seen as humble stewards of their organization’s resources.
Skill Building Progression

Unconscious Incompetence

Conscious Incompetence

Conscious Competence

Unconscious Competence
Individual Exercise

Individually think and write down answers about the following 3 questions as they relate to yourself and becoming a kaizen leader.

• What did I do lately?
• What am I going to work on next?
• What’s blocking me?

Be prepared to share your thoughts (if you would like)
Kanban Can!

We don’t “Do Kanban” we “Become Kanban”

Kanban is a journey, not a destination
Summary
This is Your Brain on Kanban
Key Takeaways

• Kanban is from a family of “pull” systems
• Kanban underpins the kaizen approach to continuous improvement
• Kanban starts with existing processes
• Visual management is a key aspect
• Kanban can be used in a variety of situations: Scrum, Waterfall, Services, etc.
• Kanban seeks a smooth, continuous flow of work
• Cumulative diagrams are key to managing Kanban systems
Kanban References

Kanban Successful Evolutionary Change for Your Technology Business by David J. Anderson

Kanban from the Inside by Mike Burrows
Changing the World

Kid President
Appendix
## Skeleton Board

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Board for Queue Exercise

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze</th>
<th>Develop</th>
<th>Accept (3)</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Analyze (3)</td>
<td>Dev (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ready for Dev (5)</td>
<td>Ready for Accept (5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Blank Board

<table>
<thead>
<tr>
<th>Ready (5)</th>
<th>Analyze (3)</th>
<th>Develop (5)</th>
<th>Accept (3)</th>
<th>Ready for Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Analyze</td>
<td>Dev</td>
<td>Ready for Accept</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ready for Dev</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bug</td>
<td>Criteria</td>
<td>Criteria</td>
<td>Criteria</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Design Complete</td>
<td>• Code Complete</td>
<td>• Acceptance Tests Green</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Test Case Examples Done</td>
<td>• Source Checked In</td>
<td>• Manual Testing Okay</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• UIX Input Ready</td>
<td>• Unit Tests Green</td>
<td>• PO Acceptance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Code Complete</td>
<td>• Build Successful</td>
<td>• Doco Complete</td>
<td></td>
</tr>
</tbody>
</table>
# Picture Catalog

<table>
<thead>
<tr>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
<th>Round 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Mail Box</td>
<td>23. Stop Sign</td>
<td>43. Square</td>
<td>63. Tall Building</td>
</tr>
<tr>
<td>4. Circle</td>
<td>24. Stick Woman</td>
<td>44. Football</td>
<td>64. Rain</td>
</tr>
<tr>
<td>10. Dog</td>
<td>30. Guitar</td>
<td>50. Key</td>
<td>70. Heart</td>
</tr>
<tr>
<td>12. Fish</td>
<td>32. Door</td>
<td>52. Graph</td>
<td>72. Keyboard</td>
</tr>
<tr>
<td>13. Hat</td>
<td>33. Box</td>
<td>53. Arrow</td>
<td>73. Funnel</td>
</tr>
<tr>
<td>14. Stick Man</td>
<td>34. Stick Dog</td>
<td>54. Stick Cat</td>
<td>74. 4 Leaf Clover</td>
</tr>
<tr>
<td>15. Hand</td>
<td>35. Road</td>
<td>55. Bicycle</td>
<td>75. Eyes</td>
</tr>
</tbody>
</table>
Recipe for Success

1. Focus on Quality
   - Under manager's control
   - Excessive defects are biggest waste in dev.

2. Reduce Work-in-Progress
   - Direct correlation between WIP and Lead Time and Lead Time to lower Quality

3. Deliver Often
   - Builds Trust with external teams
   - High quality code, delivered often

4. Balance Demand Against Capability
   - Set rate to accept new requirements based on rate to deliver working code

5. Prioritize
   - Work on priority once first 3 steps are implemented
   - Requires Product Owner to change behavior

6. Attack Sources of Variability
   - Variability results in increased WIP and Lead Times
   - Topic in statistical process control/queuing theory

David J. Anderson, Kanban
Successful Practices

• Train your team
• Demos (set regular times)
• Release Planning
• Scrum Teams
• Retrospectives
• Manage Product Backlog Collaboratively across organization
• Encourage culture that welcomes risk and innovation
Team Levels of Maturity

- **Forming**
  - Excitement
  - Anticipation
  - Anxiety
  - Optimism

- **Storming**
  - Reality sets in
  - Frustration
  - Dissatisfaction
  - Adjustment anxiety

- **Norming**
  - Shared goals
  - Team cohesion
  - Coping
  - Acceptance

- **Performing**
  - Teamwork
  - Cohesiveness
  - Leadership
  - Performance

- **Adjourning**
  - Separation anxiety
  - Crisis
  - Dissatisfaction
  - Negativity

**Next Steps**
- Options explored
- Skilled
- Anticipation
- Excitement
Kotter’s 8-Step Change Management Framework

1. **Increase Urgency**
   - Motivate, Inspire, Make Real

2. **Form a Guiding Coalition**
   - Right people at right time

3. **Create a Vision**
   - Paint picture of future

4. **Communicate for Buy-In**
   - Involve, Walk the talk

5. **Empower Action**
   - Remove Obstacles

6. **Create Short-Term Wins**
   - Bite sized chunks

7. **Consolidate Improvements**
   - Focus on ongoing change

8. **Institutionalize New Approaches**
   - Make change stick