Slide: Kanban

Roughly translates to Card and Signal

Also heard it as “sign”

Japanese term

Slide: Toyota

Part of Lean Manufacturing

Started with the Toyota Production System, brought to software by Mary and Tom Poppendieck through their work at 3M

Two books, second one seems to be the one recommended the most: Implementing Lean Software Development

Books: Toyota Production System, and the Toyota Way

7 principles of Lean Software Dev

1. Eliminate Waste
2. Build Quality in
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimize the Whole

Slide: Pull slide

Production determined by actual demand

If a bucket is full, can’t push more into it – from inventory in manufacturing

Slide: Waterfall

Long Feedback Loop

Testing well after development puts devs at a disadvantage due to their natural ADD

Scrum addresses to the iteration, 1 or 2 weeks

Slide: Calendar (IDD)

Iteration Driven Design

IDD – Scrum gone bad

Crank out dev in two week cycles, what about testing? Same 2 weeks?

Slide: Ambulance Parking

No Prioritization

No priority set

Constant priority switching

Huge back log of all high priority items

THRASHING

Slide: Reduce WIP

Reduce Work-in-Progress

Reduce multi-tasking

Reduce backlog to workable size

Slide: Balance

Balance Capacity against demand

Match work to team

Give realistic expectations

Slide: Fastpass (Prioritization)

Prioritize

The Never Bin

Slide: Focus on team

Focus on the team

Team increases cycle time and thru-put

Team works items through the queue together (Queue Limits)

Slide: Question Mark

Determine your process

Question yourself

Deeper level would be value stream mapping

High level determine -> dev, test, qa, backlog, blocked, etc

Slide: Simple Board

Scrumban – Corey Ladas

Simple first step

Slide: MMF

One Piece Flow

Minimal Marketable Feature

Slide: Big Mac Tasks

Steps of a big mac

1. Fry the two all beef patties
2. Get the sesame seed bun
3. Add the lettuce and pickles to sesame seed bun
4. Add the now fried two all beef patties
5. Put cheese on the two all beef patties
6. Blat on the special sauce
7. Sprinkle on Onions
8. Assemble sections
9. DONE!

Slide: Big Mac Kanban

Slide: Queue Limits (Weight Limit Sign)

Queue limits or WIP Limits

Queue limits for Big Mac example – 1 fry cook can make 6 patties, bun guy can only get two sets of buns ready at a time. Max Big Mac in process is therefore 2.

Queue limits for Flitter – size of team, tasks each member has to do, determine capacity

Can’t add to WIP, then swarm a feature, help in a different capacity, reduce a bottleneck

Slide: Trigger Points

Trigger points

When an item in a bucket triggers another move

Big Mac – Patties on the buns will trigger adding the cheese, then assembly

Flitter – Two items in the Dev Complete will trigger testing to start on one of them

Slide: Cycle Time

Cycle Time – “Disney Line Planning”

Time to complete work in progress: From dev through demo

Keep it balanced: don’t want 2 day dev cycle time and a 6 day testing cycle time.

Goal is to decrease cycle time.

Gain cadence, reduce waste

Different cycles for different MMFs possible. May want a S, M, L cycle estimate per MMF

Slide: Throughput

Throughput

Output over a given time

WIP/Cycle time

Allows forecasting

Slide: Incremental Planning

Incremental planning…breakout the MMF as it goes into the backlog

Decouple planning, dev, review, and release – Each can have its own cadence

Plan for it as you need it, when you need it

Leads to Late binding tasks -> Assign to the dev with the capacity at the time the task goes into WIP

Relies on the team aspect, utilize the team capacity, not just the individuals

Slide: RL Board

RL Sample board

Maintenance situation

Tried Scrummish approach with iterations, etc. Didn’t work out as tickets piled up, and focus was constantly changing

Cycle time varied from a few hours to a few days per ticket – not all tickets are created equal

Slide: MVC Board

Slide: Web Forms Board

Slide: Visualization

Visualization

The Big Board

Similar to iteration sheets, it’s all right there in front of you

Works great for co-located teams. Distributed teams takes a bit more work…some tools are in development now.

Slide: Tight feedback loop

Tight Feedback Loop

Testing done not long after dev is complete, quick feed back while task is still fresh with the developer

No waiting weeks at a time, or a whole iteration for feed back

Slide: Reduce Waste

Reduce waste

The quick feedback

The prioritization

The manageable backlog

Eliminate thrashing

Slide: Flow

Flow

Get a cadence

Gain momentum as tasks start to move (Cite RL example)

Slide: Happy Team

Happy Team

Happy devs, they’re cranking out good work

Happy client, they’re getting more bang for the buck

Happy users, they’re getting features quickly and more often