



Podcast in the 5th season
Moving to “Phase 2”

SPC, KSD, KSI certified

Lifelong coach, mentor and facilitator

Telcom, business process modeling,
enterprise agile solution delivery and
management...on and on...

1. Basics of the Agile Manifesto and Scrum Values
2. Fundamentals of agile practices
3. Why agile implementations fail and tips for successful adoption

A better
explanation to
help your journey



Quick poll:

0 = What are you talking about?

5 = I practice agility every day and I can teach it to the newbies

De-program for a moment and take a journey full of new thinking patterns

A better
explanation to
help your journey



So...Why Agile Development?

- 1. Faster, more frequent feedback loops with the customer**
- 2. Predictability and cadence in development and delivery**
- 3. Built-in quality allows for less rework and waste**



Basics of the Agile Manifesto and Scrum Values

The basis of the mindset



The 5 values of Scrum

1. **Commitment**
2. **Courage**
3. **Focus**
4. **Openness**
5. **Respect**



The 4 values of the Agile Manifesto

1. Individuals and Interactions Over Processes and Tools
2. Working Software Over Comprehensive Documentation
3. Customer Collaboration Over Contract Negotiation
4. Responding to Change Over Following a Plan



We VALUE...

Something

OVER

Something else



We VALUE... Individuals and Interactions Over Processes and Tools

Nobody said we don't value processes and tools.

We value people and our transactions, together, OVER processes and tools.

As we journey through the teamwork needed to get things done, we know that building trust is one important element to accomplishing things together.

Fundamentally, the way to correlate this into scrum is making sure that your collaboration is productive and valuable.

We don't want to check a box. We want to make sure we are delivering value.

commitment, courage, focus, openness, and respect



We value... Working Software Over Comprehensive Documentation

I don't have to tell you the history of how we got to this place.

TONS of documentation and hand-offs were at the core of all these requirements.

It was a valiant effort to attempt to deliver what the customer was asking for, but it typically failed due to re-work, waste and even obsolete functionality! That is, even if it made it to production...

Just because we practice agility, it doesn't mean we don't write anything down. It doesn't mean we just 'put a bunch of stickies on a whiteboard or wall' either.

We value documentation, but we certainly value working software more.

commitment, courage, focus, openness, and respect



We value... Customer Collaboration Over Contract Negotiation

Let's get the context behind 'contract negotiation'.

Execs or business folks work with another silo and pigeon-hole them to a strict timeline and budget.

They walk away with a contract in hand that is signed and agree to the terms of the delivery, the how and the what.

The Agile Manifesto indicates that the customer, the user, is engaged and collaborates while we are building it.

Inspecting and adapting and gaining consensus that we are building the right thing is paramount to our methods and mindset.

commitment, courage, focus, openness, and respect



We value...

Responding to Change

Over

Following a Plan

Again, context is important. There are so many reasons why this value was established.

We are not in the business of doing exactly what the 147 page 'quick guide' prescribes.

We are in the business of continual ideation, elaboration, construction and transition.

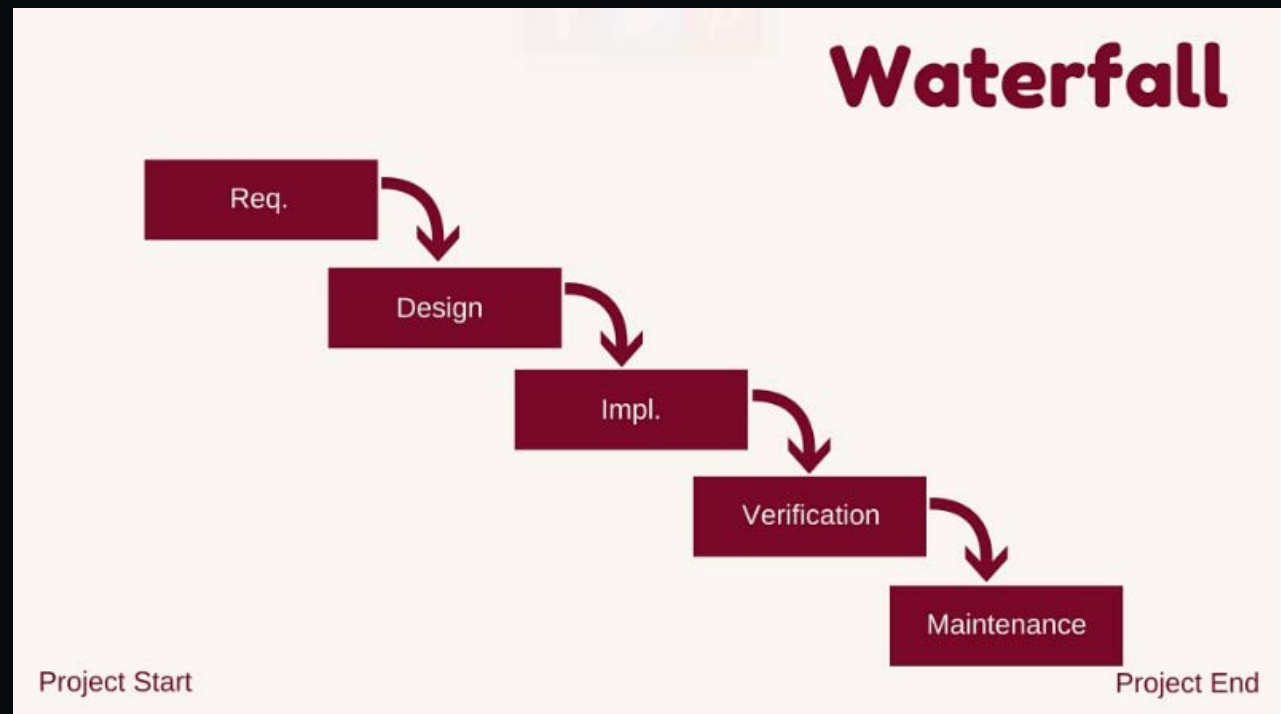
We use the other values as our guide along with this one to get delivery optimized with value and quality.

Frequent demos, smaller chunks of work and built-in quality check points make up the journey of the delivery.

commitment, courage, focus, openness, and respect



Have you ever had a long-term plan?



Have you ever considered breaking it into smaller chunks?



Sounds great!

So how do we start?

1. Recognize some **fundamental practices** that are essential to **agility**
2. *Start where you are!*
3. Don't bail on the first fail *learning experience*



Fundamental Practices are essential to agility

- ❑ **Visualize Work of Teams and Teams of Teams**
- ❑ **Limit Work In Progress**
- ❑ **Manage, Measure and Grow**
- ❑ **Working Agreements**
- ❑ **Feedback Loops**
- ❑ **Relentless Improvement**



Visualizing work

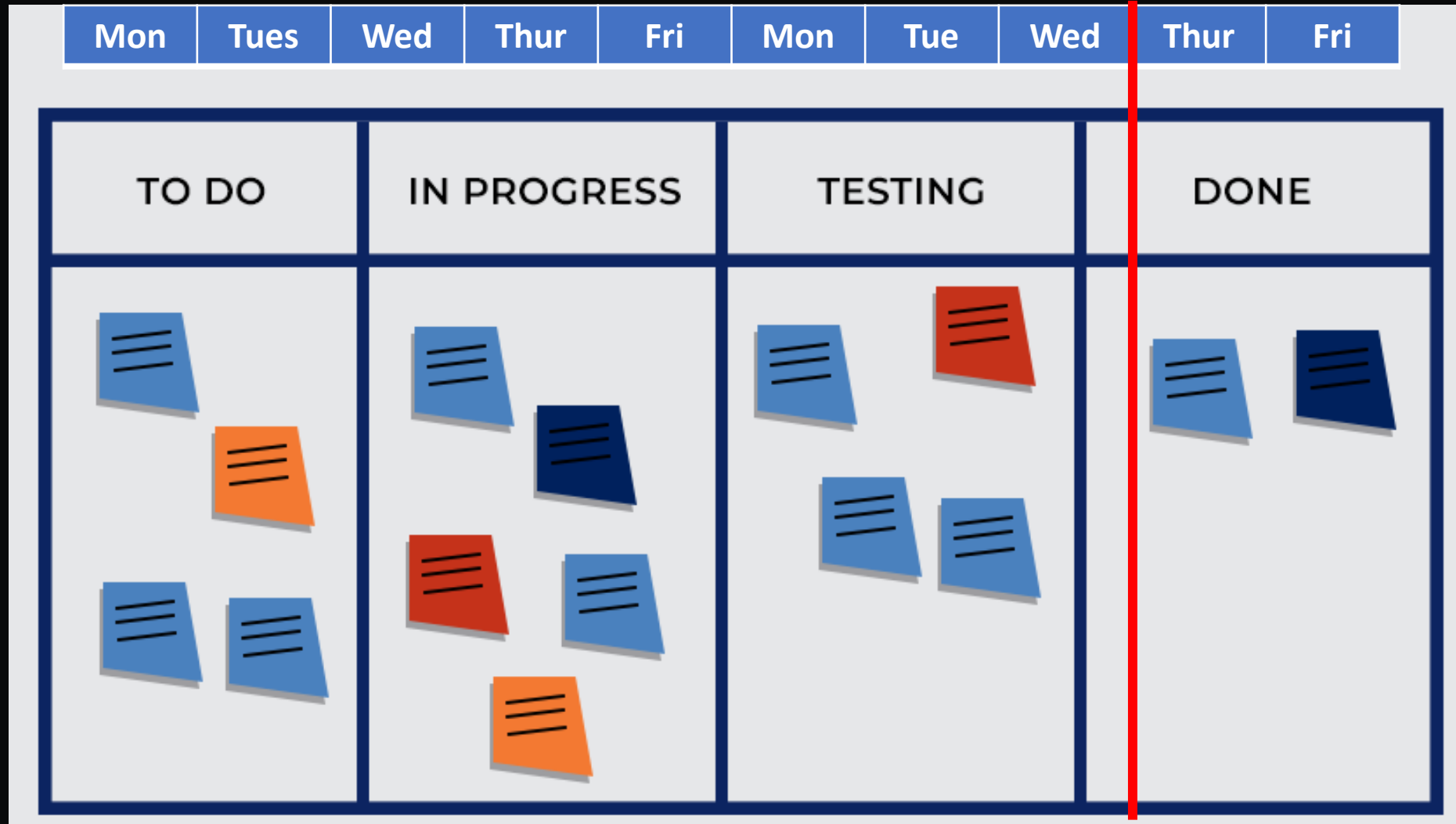
What are we working on?



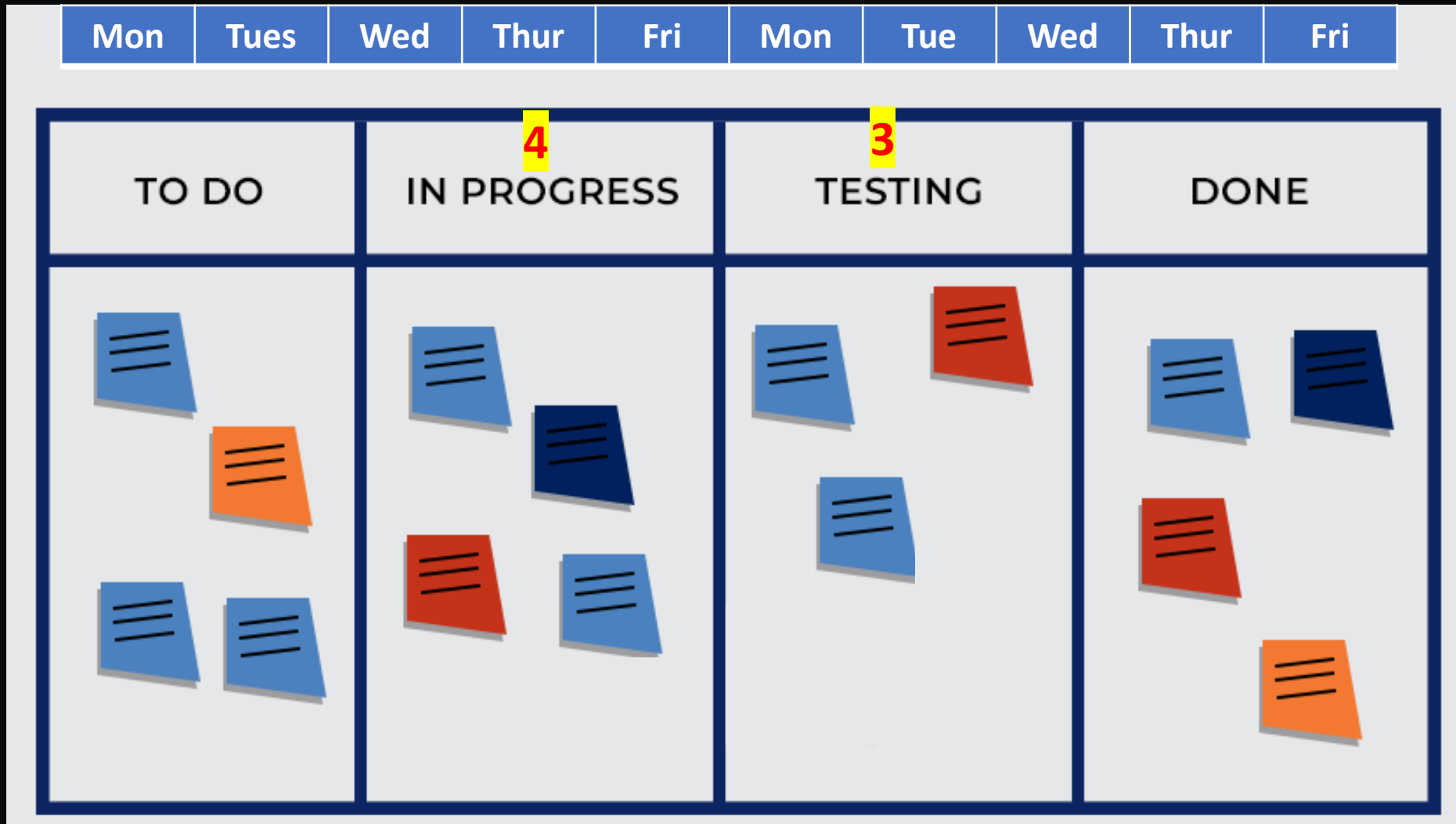
Visualizing work



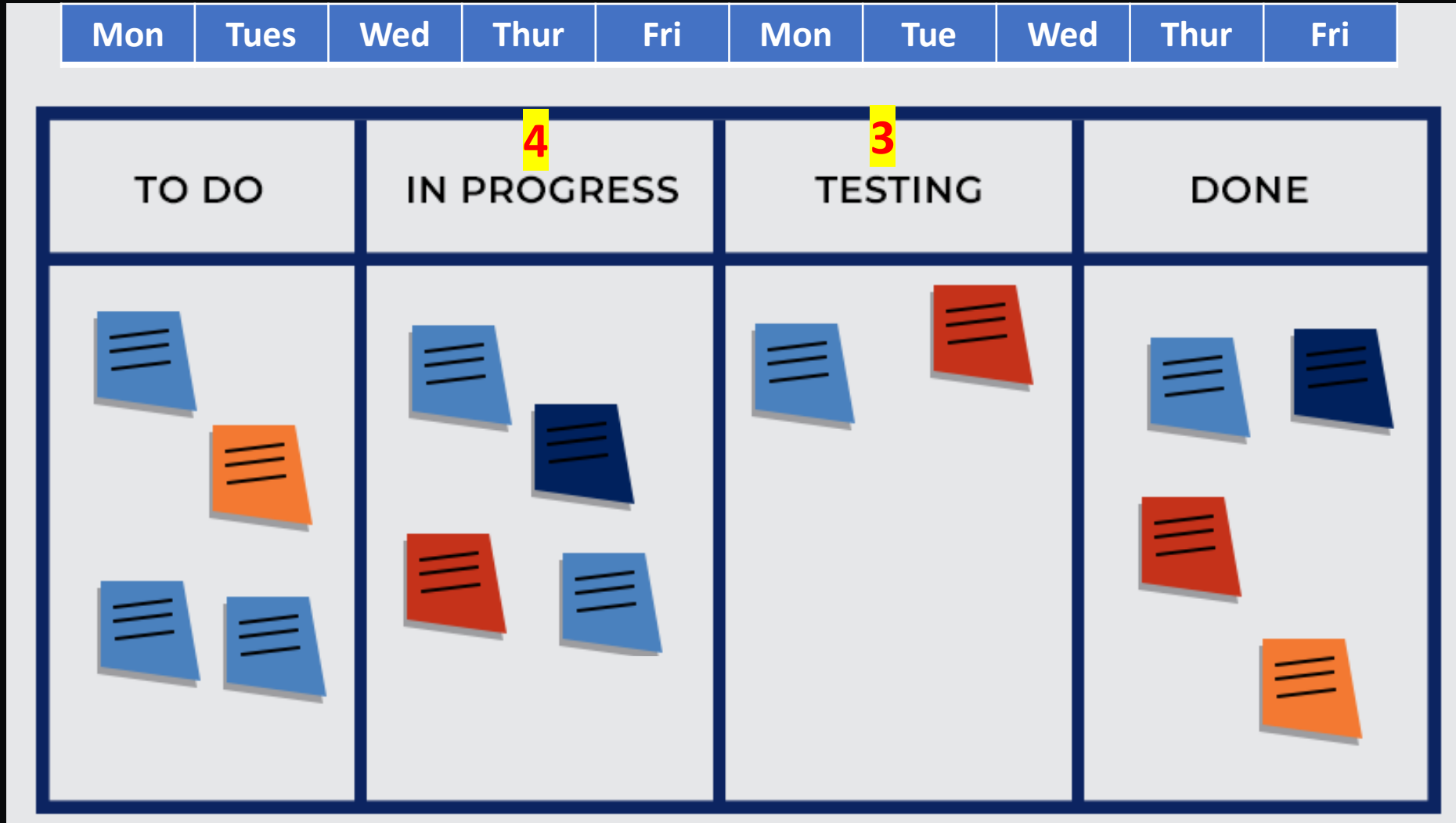
Visualizing work



Visualizing work



Visualizing work



Manage, Measure and Grow

Flow Metric	Problems Surfaced	Flow Accelerator to Remediate
Flow Distribution – The portion of each backlog item type	Too much focus on business features leading to solution health degradation, slowing development	<ul style="list-style-type: none"> 4. Get fast feedback 6. Manage queue lengths
Flow Velocity – number of items completed in a given time	<p>Underlying problems with productivity</p> <p>Unpredictable velocity from one time period to the next</p>	<ul style="list-style-type: none"> 5. Work in small batches 7. Optimize 'time in zone'
Flow Time – time work spends in the workflow	Slow time to market causing customer to wait and incurring a cost of delay	<ul style="list-style-type: none"> 1. Visualize and limit WIP 3. Minimize handoffs and dependencies 5. Work in small batches 8. Remediate legacy policies and practices
Flow Load – total work in process	Excess work in process leading to increased flow time as queues build up in the system	<ul style="list-style-type: none"> 1. Visualize and limit WIP 6. Manage queue lengths
Flow Efficiency – ration of active time to total time	Large amounts of waste in the system along with bottlenecks and delays that need addressing	<ul style="list-style-type: none"> 2. Address bottlenecks 3. Minimize handoffs and dependencies 8. Remediate legacy policies and practices
Flow Predictability – planned vs actual business value delivered	Low or erratic predictability highlights underlying problems in technology, planning, or organization performance that need addressing	<ul style="list-style-type: none"> 4. Get fast feedback 5. Work in small batches



Working Agreements

- **General**
- **Non-Team meetings and work sessions**
- **Team Rhythm, Cadence and Ceremonies**
- **Workflow**
- **Decision-making**
- **Communication, Resources etc**



General Agreements

What applies to all our work?

What are we going to hold each other accountable to?

What are some basic agreements here?

ELMO –
Enough, let's
move on

Respect,
inclusion etc

Let team (or one
team member?)
when you plan to
be OOO

WAM –
Wait a minute
("I don't get it")

Don't interrupt
others ; Stay on
mute when not
speaking

Office
hours

Leave for the
day after demo

The buzz word is
'pineapple' means
I'm overwhelmed
and need help

If a dispute takes
place, we bring
in.. _____

Unit / Peer /
Pair approach?

When team member
gets pulled into
'emergency', then
we will _____

Quality / Testing
methods (TDD, BDD,
Unit/UAT etc)



Non-Team Meetings and Work Sessions

Define parameters around NON-TEAM meetings / sessions

What is the expectation

Who is expected to join which types?

Meeting
URLs are
posted in _____

_____ won't
attend
"Meetings" with
no value to WIP

Video required
for xxx type
sessions

No work
sessions
scheduled
during _____

_____ must
attend
"Meetings" not
related to WIP

Video NOT
required for
yyy sessions



Team Rhythm, Cadence & Ceremonies

What rhythm do we expect?

What is the cadence we expect?

How do we know we are learning?

Plan TEAM Events / Ceremonies here

Open
availability
forum between
9-10AM on Tues
and Thurs

2-week iteration
of continual
flow / Sprint
timebox

Demo times and
link

Standup or
"touch-point"
frequency

_____ type
sessions are
team-only

Team work
sessions will be
on _____ open
between x-y

Team planning
times, locations
and durations

Retro / planning
- who is required,
when, etc



Workflow

How will we inspect and adapt?

How do we ensure flow?

How do we define classes of service, priority and systems (boards)?

What are some rules of entry/exit/demo etc?

Definition of Done is found here (LINK)

Definition of ready met before / after 'TO DO'?

Team will 'swarm' based on class of service

Demo to PO and peer before moving to 'done' (Like "Review" lane)

Definition of ready met before / after 'TO DO'?

TEAM members can only can "pull work"

Swarmed work requires pilot and co-pilot

Team will revisit definition of done in each retro

Team can / can't pull work in under x,y,z conditions

If unable to "pull" work...?

Class of service in WIP wins priority...

Pull down new code to IDE every morning



Decision-Making

What is the method to get approvals?

What types of decisions need to be approved? (high level)

NFR / Pen / Perf go/no-go

Who accepts or rejects 'done' work?

Who can make the call on 'dirty pushing'?

PO is the only team member who can add to 'TO DO' lane

Iteration goals are set by the team

Define the metrics here, too, like cycle-time, lead-time, velocity

Compliance / DBA / Architect etc must approve task types with _____

Program level agreements govern product backlog and system access?

At what point in the process (program Kanban, anyone?) should we dive deep into the next thing?



Resources, Info & Communication

Who are we, what do we do and our purpose?

Who and what do we need for support?

How do we manage conflicts when one is not available?

How is the best way for us to communicate as a team?

Where do we put our information and who puts it there?

What roles do we have now?

Team will communicate via...

When work is 'done', we will tell ___ by ___

List environments and entry/exit criteria for each (maybe based on work type)

Do we have a backup in case of a vacation, emergency, conflict in schedule?

What is our team vision, motto, name, charter etc?

Team agreements will be stored where?

Institutional knowledge, technical, testing gaps etc

Who are the strategic and tactical partners / PoC's for work, architectures, functional etc?

*Did we start with any gaps?
What role will we soon need?*

_____ must provide notes on _____



Team name:

Team Mission Statement:

Team roles and members:

Working Agreements

General:

- Treat everyone with respect
- Let everyone finish their statement
- Mute yourself when not speaking
- “WAM” = Wait a Minute!
- “ELMO” = Enough, Let’s Move On!
- “Pineapple” = I need to take a break!

Rhythm, Cadence & Ceremonies:

Non-Team meetings:

Workflow:

Decision-Making & Risk Management:

Resources, Info and Communication:



Feedback Loops

- ❑ **Daily standup / Team Sync**
- ❑ **Scrum of Scrums / Coach Sync**
- ❑ **Inspect and Adapt**
- ❑ **PI Planning / SDA planning**
- ❑ **Sprint Review / Demo / Iteration Review**
- ❑ **Iteration / Sprint Planning**
- ❑ **Iteration / Sprint Retrospective**
- ❑ **ShareCare Time**
- ❑ **Solicited / non-solicited feedback from customers!**



Relentless improvement

- ✓ **Plan**
- ✓ **Do**
- ✓ **Check**
- ✓ **Act**



Why do agile implementations fail?

- ≠ **Lack of executive commitment**
- ≠ **Lack of realizing value**
- ≠ **Lack of support from mid-management**
- ≠ **Lack of training/certifications**
- ≠ **Lack of belief in the 'system'**
- ≠ **Lack of patience to let it unfold**



Reason #1: Expectations are supported by myths

5 popular myths about “Agile”

1. We are gonna “DO” Agile!
2. Get everyone trained and certified and we are golden!
3. We can't ‘do agile’ because it doesn't support needed documentation!
4. If we ‘do agile’, we will be faster!
5. Agile is a free-for-all with no planning, no discipline and no accountability!

5 realities of agility

1. Agility is a philosophy and a mindset
2. Certifications combined with support, adoption and implementation create value
3. Agile practices require documented working agreements, solutions and guidance
4. Agility allows for frequent feedback loops that build in quality (PDCA)
5. Agile practices require support for self-organization, trust and respect



Reason #2: The process becomes the product

The process can't be the product.

The process should improve the product.

1. Traditional risk management and project offices needs to ensure governance
2. Overwhelming communication, documentation and elaboration policies and procedures take the place of collaboration, trust and transparency
3. Too many people in meetings yields too many 'one-off' conversations without the right people
4. Time wasted on events, reporting and processes could be spent working on items to deliver to customers
5. 147 people to approve a solution OR 147 solutions to approve indicates a lack of priority



Reason #3: We don't like change and we love control

People like change. It is the process of change that we don't like.

1. If 'command and control' drives the bus, empowerment takes a back seat
2. Shared resources with conflicting priorities makes for over-allocation, context-switching and a loss of purpose in their role
3. Silos and groups struggle to organize around value (fund value streams over projects)
4. If we are too concerned with reporting structures, we are missing the point
5. Hire good people, lead them to own the work, challenge them to deliver more frequently with quality and support them when they are in need



Reason #4: Inconsistent messaging

House rules in Monopoly are great if we agree before we play!

1. Domain language is crucial
2. Agree to the definitions of product owner, daily stand up, etc and write it up in a working agreement
3. Bending definitions, abusing language, inconsistent staff utilization and making up rules out of 'convenience' sends a terrible message from leadership
4. Your team members, associates and those who went through the training want you to 'do what is in the book'
5. Acknowledgement of trust and self-organization requires us to relinquish control or a budget item



Reason #5: We struggle with measuring success

We should be measured on delivering working software with quality in order to test our hypotheses with our customers

1. If we “Can’t do this”, then let the team figure out how...and make the decision to support it, improve it, reject it or table it
2. Measure what you manage and manage what you measure.
3. Rather than focusing on status reports, focus on removing impediments.
4. Conflicting reporting measurements and metrics within the organization requires a great deal of attention on customizations and less focus on the customer
5. Focus on team-centric metrics that move the needle for customer-centric metrics



Q & A!

Email:

scott@planetproductowner.org

YouTube:

@planetproductowner

Podcast: (most ANY platform!)

Planet Product Owner

