Project Management at NetSuite
Jose Solorio
Agenda

- Background
- NetSuite
- Project Management
- Why Have a Methodology at all?
- Waterfall
- Scrum
- How to Transform a Company to Scrum
Background

Senior Project Manager & Scrum Master

6 years project management experience, specializing in:

- Agile / Scrum Methodology
- User Experience & User-Centered Design (UE/UX)
- Web-Based Applications
- Software Development

Previous companies

Certifications

UCSC Alumn

Information Systems Management, B.S., 2008

Former Information Systems Management Association – ISMA – Officer

linkedin.com/in/jsolorio

jsolorio@netsuite.com
Agenda

- Background
- NetSuite
- Project Management
- Why Have a Methodology at all?
- Waterfall
- Scrum
- How to Transform a Company to Scrum
Company

- Founded in 1998 by Evan Goldberg, CTO, with financial backing of Larry Ellison (Founder and CEO of Oracle)

- Originally named NetLedger

- December 2007: Publicly Traded NYSE “N”

- NetSuite.com
Locations

US
- San Mateo, CA (HQ)
- Paso Robles, CA
- Denver, CO
- Boston, MA
- Austin TX
- New York, NY
- Las Vegas, NV

Intl.
- Australia
- Canada
- Czech Republic
- Singapore
- United Kingdom
- Japan
- Philippines
- Hong Kong
- Uruguay
What We Do

NetSuite is the world’s #1 cloud business management suite

Cloud: Hosted and maintained online

Our SaaS (Software-as-a-Service) products allow businesses to run their entire operations online using a single tool

- Financials
- Accounting and Operations
- Payroll and Time Tracking of Employees
- Order Management
- E-Commerce
- Business Intelligence
Agenda

- Background
- NetSuite
- Project Management
- Why Have a Methodology at all?
- Waterfall
- Scrum
- How to Transform a Company to Scrum
What is a Project?

- A temporary endeavour with a beginning and an end that aims to create a product, service, or result.

- Time, Scope, and Resources are usually known as the 3 Constraints. One constraint cannot be changed without affecting the others.

What is Project Management?

- My definition: Applying **knowledge**, **tools**, and **experience** to successfully guide a team towards producing a product, service, or result
  - **Knowledge**: Solid understanding of the composition of the work
  - **Tools**: Using applications or software to measure our progress
  - **Experience**: Having seen many types of situations and knowing what to do in each case

- Various ways to manage a project, called Methodologies
  - Traditional (Waterfall)
  - Agile (Iterative)

- Methodologies are processes that teams follow and define what to do, the sequence to do it in, etc.
Project Management

The Role of the Project Manager

- **Planning**: Working with team members to breakdown what needs to be worked on first, second, etc
- **Tracking**: Having a complete understanding of the health of the project at any one time
- **Communicating**: Reporting the project health to audiences of various kinds
- **Collaborating**: Reaching out to the right individuals to get the next task done
- **Coordinating**: Staying one step ahead of the team by setting up discussions and removing impediments for them
What Specialization Areas of Project Management Exist?

- Application Development
- Hardware Deployment (Servers, switches, networking gear)
- Security Systems & Software
- Multi-Organizational Initiatives
- Backend-Specific coding (Java, PERL, Python, C)
- User Experience Project Manager
- IT Project Manager
- Security Project Manager
- Cross-Functional Project Manager
Agenda

- Background
- NetSuite
- Project Management
- Why Have a Methodology at all?
- Waterfall
- Scrum
- How to Transform a Company to Scrum
Lets backtrack a little, why have a methodology at all?

Companies need to make decisions on where to spend money.

- “Which direction do we want the product to go? Value versus cost. “
Let's backtrack a little, why have a methodology at all?

These are big decisions, and in order to make those decisions, organizations use plans to chart a roadmap of where the company needs to go, how capital will be spent, and where commitments will be made.

In order to make commitments, managers rely on plans that contain estimates that try to answer the age-old question Value vs Cost: “If I have a team that cost this much, how long will it take them to deliver the product (value)?”

Having a plan allows → Project commitment to be made → Allow companies make decisions on which direction to go.
Lets backtrack a little, why have a methodology at all?

- Good plans help managers make informed decisions
- But there’s a problem: Plans are always wrong because they are just that, an estimate, and not a guarantee
- Plans don’t guarantee anything, they just convey an estimate, which is just a possibility
- A possibility that a project can finish either:
  - On-Time
  - Early
  - Late
  - Or not even at all!
Let's backtrack a little, why have a methodology at all?

You’re probably thinking…

What if a team tries really, really hard to come up with a good plan, tries its best to do accurate estimates, give very precise dates, spends all this time planning it out to eliminate any risk of not finishing on time. Wouldn’t that work?
Let's backtrack a little, why have a methodology at all?

Answer 1:

Law of Diminishing Returns:

After a certain point, you’re wasting too much time thinking about the problem and not actually solving it.
Let's backtrack a little, why have a methodology at all?

Answer 2:

Let's assume you went ahead and spent all this time making elaborate, detailed plan anyway and somehow you managed to deliver the product by the exact date your plan said...

The plan still doesn't guarantee that the customers will like or use the product! What if you amazingly delivered a product that the customer did not want at all.

Here's what these guys have to say about delivering something that customers don't want:

Way to go!
Let's backtrack a little, why have a methodology at all?

- So on one hand, you have Plans that are always wrong.
- On the other, you have companies that demand accurate plans so that solid commitments can be made so that the company can move forward.

So what do you do?
- **More** Planning?
- **Less** Planning?
- **No** Planning at all?

Let's take a look at how 2 methodologies go about handling this, Waterfall and Scrum.
Agenda

- Background
- NetSuite
- Project Management
- Why Have a Methodology at all?
- Waterfall
- Scrum
- How to Transform a Company to Scrum
It’s called Waterfall because the process looks like one.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
</tr>
<tr>
<td>Coding / Implementation</td>
<td></td>
</tr>
<tr>
<td>Integration</td>
<td></td>
</tr>
<tr>
<td>Testing and QA</td>
<td></td>
</tr>
<tr>
<td>Deployment</td>
<td></td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
</tr>
</tbody>
</table>

Start → Finish
Waterfall

- Waterfall tries to break up a plan into sequential phases
- The idea is that when you finish a phase, you can be 100% sure it is done and will NEVER repeat
- The hope is that: When you finish a phase, you’ve discovered all there is to know and answered all the questions there are to answer so that you never have to go back and revisit it or do rework

KEEP CALM AND FINGERS CROSSED
But more often that not, things never go as planned.

If you are using Waterfall, you keep planning until the planning is done. It becomes a day-by-day slip, pushing everything else out.

But if you are using Scrum, you do just enough planning.
Agenda

- Background
- NetSuite
- Project Management
- Why Have a Methodology at all?
- Waterfall
- Scrum
- How to Transform a Company to Scrum
A sprint is a timeboxed period with a fixed start and finish.

A sprint has 5 steps. All steps must finish in the sprint.

Usually, organizations do 2-week sprints, but other teams have theirs be anywhere from 1wk – 4wks.

A sprint produces potentially shippable product, component, service, or result.

A sprint doesn’t make a team work faster, it allows them work at a sustainable pace on smaller pieces at a time.

The team works in parallel (Dev and QA start at the same time).
Scrum – Iterative Framework

Release 1.0

Multiple Sprints form a Release (3-6 Months)

Mobile Apps Theme

Architecture Redesign

Multiple Releases form a Theme (1-2 years)
Scrum defines 3 roles

- **Product Owner (1)**
- **Scrum Master (1)**
- **Development Team (Any**)**

The team is everyone inside the circle

** Ideal team sizes range from 5 – 9, including the Product Owner and Scrum Master**
Scrum defines 3 roles

Product Owner

- Prioritizes the team’s work

- Takes 4 things into consideration when prioritizing
  1. Financial value of having those features
  2. Cost of developing
  3. Knowledge generated by developing those features.
  4. Risk removed by developing those features
Scrum defines 3 roles

Product Owner

Product Owners prioritize based on Risk-Value

<table>
<thead>
<tr>
<th>Risk</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Avoid</td>
</tr>
<tr>
<td>Low</td>
<td>Do Last</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

Work on High-Risk, High-Value first: They eliminate the most risk out of the way.

General Rule: Work first on High-Value, but use Risk as a tie-breaker, where Risk supersedes.
Scrum defines 3 roles

Product Owner

According to Mike Cohn, the top 5 Product Owner qualities:

<table>
<thead>
<tr>
<th>Quality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available</td>
<td>Makes himself/herself available as much as possible for questions the team may have</td>
</tr>
<tr>
<td>Business-Savvy</td>
<td>Should understand the business landscape, market conditions, customers, needs</td>
</tr>
<tr>
<td>Communicative</td>
<td>Communicates and listens to user, customers, the team. Should hear recommendations.</td>
</tr>
<tr>
<td>Decisive</td>
<td>Will not try to avoid a hard decision by saying “Let me get back to you” and have the team wait</td>
</tr>
<tr>
<td>Empowered</td>
<td>If a Product Owner is constantly overruled by other higher-ups, the team will try going to those guys first next time for answers</td>
</tr>
</tbody>
</table>
Scrum defines 3 roles

Scrum Master

- Owns the Scrum process
- Removes impediments for the team
- Coaches the team on the Scrum practices
- Ensure the team doesn’t overcommit to what can be achieved
- Works with the Product Owner to ensure the Backlog is in a “Ready-State”
- Produces performance metrics
- Manages communication of project status
- Coordinates activities with other teams
- Is both a leader and someone with no authority (Nobody reports to me)
Scrum defines 3 roles

Top-6 Scrum Master Qualities

<table>
<thead>
<tr>
<th>Quality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible</td>
<td>Maximizes team output, assists the team, and provides guidance</td>
</tr>
<tr>
<td>Humble</td>
<td>They say “Look what <strong>WE</strong> accomplished”, never “look what <strong>I</strong> accomplished”. Lead by example</td>
</tr>
<tr>
<td>Collaborative</td>
<td>Encourage teams to think together to solve a problem</td>
</tr>
<tr>
<td>Committed</td>
<td>Leave no impediment unattended</td>
</tr>
<tr>
<td>Influential</td>
<td>Influences others, both in the team and outside</td>
</tr>
<tr>
<td>Knowledgeable</td>
<td>Have technical, market, and other specialized skills. Knows sufficient to lead a team</td>
</tr>
</tbody>
</table>

Credits: Mike Cohn
Scrum defines 3 roles

Development Team

- They do the work

- Can consist of not only Software Developers, but also Testers (QA), Designers, Database Administrators, Security Analysts, etc.

- Cross-Functional

- Deliver quality assured product incrementally

- Self-organizing

- Demonstrate the work to the Product Owner at the end of each sprint
So you have a Team, and you have a Process, how do you apply it?
Scenario

You’ve been called to manage a team to create the next NetSuite Mobile App using a Scrum methodology.
Sprint Planning

- Sprint Planning
- Update Product Backlog
- Sprint Work
- Sprint Review
- Sprint Retrospective

Sprint 1

Sprint X

<table>
<thead>
<tr>
<th></th>
<th>Week 1</th>
<th></th>
<th>Week 2</th>
<th></th>
<th>Week 3</th>
<th></th>
<th>Week 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monday</strong></td>
<td></td>
<td><strong>Tuesday</strong></td>
<td><strong>Wednesday</strong></td>
<td><strong>Thursday</strong></td>
<td><strong>Friday</strong></td>
<td><strong>Monday</strong></td>
<td><strong>Tuesday</strong></td>
<td><strong>Wednesday</strong></td>
</tr>
<tr>
<td>Sprint Planning</td>
<td></td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Review</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Retrospective</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
</tbody>
</table>

40
**It all starts with the Product Owner**

What would you like to be able to do?

<table>
<thead>
<tr>
<th>Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to search</td>
</tr>
<tr>
<td>Reorder</td>
</tr>
<tr>
<td>Home button</td>
</tr>
<tr>
<td>Calendar</td>
</tr>
<tr>
<td>Customers</td>
</tr>
<tr>
<td>Leads</td>
</tr>
<tr>
<td>More options</td>
</tr>
<tr>
<td>See saved searches</td>
</tr>
<tr>
<td>View Sales</td>
</tr>
<tr>
<td>View Income</td>
</tr>
<tr>
<td>View Expenses</td>
</tr>
<tr>
<td>Have report snapshots</td>
</tr>
<tr>
<td>View list</td>
</tr>
<tr>
<td>View in Landscape Mode</td>
</tr>
<tr>
<td>Settings</td>
</tr>
</tbody>
</table>
## Release 1

### Backlog

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to search</td>
</tr>
<tr>
<td>Reorder</td>
</tr>
<tr>
<td>Home button</td>
</tr>
<tr>
<td>Calendar</td>
</tr>
<tr>
<td>Customers</td>
</tr>
<tr>
<td>Leads</td>
</tr>
<tr>
<td>More options</td>
</tr>
<tr>
<td>See saved searches</td>
</tr>
<tr>
<td>View Sales</td>
</tr>
<tr>
<td>View Income</td>
</tr>
<tr>
<td>View Expenses</td>
</tr>
<tr>
<td>Have report snapshots</td>
</tr>
<tr>
<td>View list</td>
</tr>
<tr>
<td>View in Landscape Mode</td>
</tr>
<tr>
<td>Settings</td>
</tr>
</tbody>
</table>

### Sprint 1

- Ability to search
- Reorder
- View Sales

### Sprint 2

- View Income
- See saved searches

### Sprint 3

- View Expenses
- View list
- Home button
- Customers

**Highest Priority**

(High-Risk, High-Reward)

**Lowest Priority**

(Low-Risk, Low-Reward)
Sprint Work

Sprint Planning

Sprint Work

Sprint Review

Update Product Backlog

Sprint Retrospective

Sprint 1

Sprint X

Week 1 | Week 2
---|---
Monday | Sprint Planning
Tuesday | Sprint Work
Wednesday | Sprint Work
Thursday | Sprint Work
Friday | Sprint Work
Monday | Sprint Work
Tuesday | Sprint Work
Wednesday | Sprint Work
Thursday | Sprint Work
Friday | Sprint Review

Update Product Backlog

Sprint Retrospective
### Sprint Review, Retrospective, Update Product Backlog

#### Sprint 1

- **Sprint Planning**
- **Update Product Backlog**
- **Sprint Work**
- **Sprint Retrospective**
- **Sprint Review**

#### Sprint X

<table>
<thead>
<tr>
<th>Week 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Week 2</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Monday</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Planning</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tuesday</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wednesday</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Thursday</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Friday</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Sprint Review

- **Friday**: Sprint Review

#### Sprint Retrospective

- **Friday**: Sprint Retrospective

#### Update Product Backlog

- **Friday**: Update Product Backlog
Sprint Review

Sprint 1
- Ability to search
- Reorder
- View Sales

NetSuite Interface with indicators showing an increase of 446.6% in Sales.
“OK, now that we all agree, let’s all go back to our desks and discuss why this won’t work.”
# Update Product Backlog

## Backlog

| Calendar |
| Leads |
| More options |
| Have report snapshots |
| View in Landscape Mode |
| Settings |

## Sprint 1
- Ability to search
- Reorder ✓
- View Sales ✓

## Sprint 2
- View Income
- See saved searches
- Customers

## Sprint 3
- View Expenses
- View list
- Home button
- Customers
- More options

---

Moved **More Options** from Backlog Sprint 3

Moved **Customers** from Sprint 3 to Sprint 2
Sprint 2 Planning

We start the cycle all over again

Sprint Planning

Update Product Backlog

Sprint Work

Sprint Review

Sprint Retrospective

Week 1

<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint Planning</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
</tbody>
</table>

Week 2

<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Work</td>
<td>Sprint Review</td>
</tr>
</tbody>
</table>

Update Product Backlog
Sprint 2

Fast-Forward
Sprint 2

View Income  ✓
See saved searches  ✓
Customers  ✓
Sprint 3

View Expenses
View list
Home button
More options
Scrum – Iterative Framework

Release 1.0

Mobile Apps Theme
Finished Product After Theme Is Complete
Agenda

- Background
- NetSuite
- Project Management
- Why Have a Methodology at all?
- Waterfall
- Scrum
- How to Transition a Company to Scrum
To transition, a company needs to be desiring and willing to do so. **If that’s not in place, forget it!**

Transitioning a company to Scrum is not easy; it comes with its challenges:

- You need upper management AND lower-level support from the team members that will be participating
- Scrum needs to be tailored to fit the circumstances of the company
- Scrum will change everything about a team member’s typical workday
- Scrum is dramatically different
- With Scrum, previous behaviors have to be unlearned
- You **will** encounter resistance from people. **WILL.**
But the effort is worthwhile. Should it succeed, you will see:

- Higher productivity
- Eliminated waste
- Teams less likely to build functionality that is no longer needed
- Improved employee engagement and job satisfaction
- Faster time to market
- Higher quality code
- Bugs fixed in the same sprint
- Improved customer satisfaction
To convince an adoption of Scrum, follow 5 steps: **ADAPT**

- **A**wareness that current process is not working
- **D**esire to adopt Scrum to address existing issues
- **A**bility to succeed with Scrum
- **P**romotion of Scrum through sharing of other successful experiences
- **T**ransfer the implications of using Scrum through the company
To convince an adoption of Scrum, follow 5 steps: ADAPT

- Awareness that current process is not working

There is usually a lag between the time when change is need and when we become aware that it is needed

Sometimes, the need for change is only obvious to certain departments

Sell the problem (We are missing our deadlines), not the solution (We need a new process)

How? Use metrics / data as reinforcements for reasons to change. Examples:

- Satisfaction surveys
- Revenue per employee
- Employee turnover
To convince an adoption of Scrum, follow 5 steps: ADAPT

- Desire to adopt Scrum to address existing issues

To increase desire to adopt Scrum:

- Communicate that there’s a better way. Be very specific
- Create a sense of urgency
- Build momentum. Focus on those who are enthusiastic about the transition
- Do a 2-3 month trial and get those teams to succeed

Try to get one of the senior managers or engineers on your side, this will help convince others to try
To convince an adoption of Scrum, follow 5 steps: ADAPT

- Ability to succeed with Scrum

Provide coaching and training

Hold individuals accountable for applying those skills that the organization is paying to acquire

Cross-pollinate: Invite members from other scrum teams to join yours to share techniques

Set reasonable targets
To convince an adoption of Scrum, follow 5 steps: ADAPT

- Promotion of Scrum through sharing of other successful experiences

Reinforce agile behavior on existing teams by spreading good news achieved

When other see it, they will want to do it too

Creating interest in people outside of your team will decrease your chances of getting dragged back away from Agile or having the effort implode

Let others attend your scrums and learn by joining as participants
To convince an adoption of Scrum, follow 5 steps: ADAPT
- Transfer the implications of using Scrum through the company

A Scrum team needs to be interoperable with the rest of the company. It can’t remain agile on its own permanently unless if the implications of scrum are transferred to the other departments too

You don’t have to convert every other department (HR, Facilities, Marketing, Finance, etc.) to scrum. They just need to make a few small changes to be compatible
There are 7 different approaches to convert a team to Scrum:

<table>
<thead>
<tr>
<th>Approach</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Small</td>
<td>Focus on converting 1-3 teams and get them to be successful</td>
</tr>
<tr>
<td>All-In</td>
<td>Convert the entire company all at once</td>
</tr>
<tr>
<td>Public Display of Agility</td>
<td>Announce to the company that Scrum transitioning is taking place, make a big deal out of it</td>
</tr>
<tr>
<td>Public Display of Stealth</td>
<td>Only the scrum team being converted knows, nobody else</td>
</tr>
<tr>
<td>Split-and-Seed</td>
<td>After you convert a team, you dissolve the team members into other teams that have not yet transitioned.</td>
</tr>
<tr>
<td>Grow-and-Split</td>
<td>You grow a team to be large enough until it can then be split into 2 smaller ones</td>
</tr>
<tr>
<td>Coach</td>
<td>You bring in an agile coach and move the coach from team to team. Teams remain intact, but the coach migrates from team to team</td>
</tr>
</tbody>
</table>
Other Considerations

- Be careful how you seed people. I.e. If you pair up C programmers with JavaScript programmers in situations where they can’t code things together, it doesn’t make sense.

- Pairings that won’t work should be avoided

- If everyone on a team is excited about Scrum except for one person who is very resistant, if possible, try to move that person out of the team

- Pessimism and bad behavior from one person can ruin everyone else’s effort and desire to adopt Scrum

- Resistance is not a problem, it is an indication that someone needs help understanding Scrum
Recap

- We have the 5 ADAPT steps
- We have 7 approaches to convert a team to Scrum
- Which projects do you proceed to convert first?
  - Everyone will be looking at the first project you convert
  - Selecting the right project and the right team is vital
Pushing towards Agility

- Once you select a project(s), a team(s), approach, and kick-off ADAPT, move quickly

- If possible, run several pilot projects so that all hopes aren’t tied up to just 1 project incase if that one fails

- Begin with Release Planning, followed by Sprint Planning and start your first sprint

- Keep an eye on the team so that nobody tries to bring with them the old habits of Waterfall into the sprint
Pushing towards Agility

Set expectations about 4 things upfront:

- **Progress**: Teams will be slow at first, but will speed up later

- **Predictability**: It will take several sprints for the team to figure out how much work they can consistently deliver per sprint (Capacity)

- **Attitudes**: Expect people (both inside the team and outside) to complain at first. Don’t give in. Make an agreement and stick to it

- **Involvement**: Ensure stakeholders understand the level of commitment needed from each team member for this to work
How to Transition a Company to Scrum

Technical practices to apply

- TDD – Test Driven Development
- Refactoring
- Collective ownership
- Continuous integration
- Pair-programming
- Testing automation across 3 layers
How to Transition a Company to Scrum

Technical practices to apply

- **TDD – Test Driven Development**
  - You first write a test that will fail. Then, you write just enough code to pass it. Then refactor (Improve code quality)

- Before, programmers would first fully code a feature, then write the test case after. Problem: Some code would still make it into the system untested

- With TDD, no code makes it into the rest of the system untested

- With TDD, less bugs, less time wasted fixing bugs
How to Transition a Company to Scrum

Technical practices to apply

- Refactoring (Extract method)
  - Changing the structure without changing the behavior of the code
  - You do this a little at a time when it won’t cost too much effort
  - Major refactoring should be its own user story in the Backlog
  - The team should decide when to do major refactoring and be very specific on exactly what will be refactored and how much
How to Transition a Company to Scrum

Technical practices to apply

- Collective ownership
  - No one should say “That code belongs to Frank, I can’t touch it”

- We all own the code

- Nobody should be allowed to specialize so much in one area of the code that only that person can work on it later

- Quality is everyone’s responsibility
Technical practices to apply

- Continuous integration
  - Integrating the code into the rest of the system as soon as it’s checked in

Scrum needs an automated test environment to deliver the best results anyway. The time it takes to setup a continuous integration server will be given back to the team in the form of saved time and saved integration issues later

- Will allow to deploy at any time

- If running a full integration takes will too long, split the tests into:
  - **Smoke test**: Quick run, after every check-in, contains a subset of the tests
  - **Full test**: Run every hour, includes all tests from all milestones
  - **Nightly build**: Run once a day
Technical practices to apply

- Pair-programming
  - 2 programmers sitting side-by-side, one keyboard, writing code. One types, the other looks over and navigates

- 2 minds working on a problem is better and quicker than one

- Can help bring the other developer up to speed on the code (2 programmers being familiar with the code instead of just 1)

- Having an extra pair of eyes can help catch more bugs
How to Transition a Company to Scrum

Technical practices to apply

- Testing automation across 3 layers
  - UI – User Interface
  - Service Layer
  - Unit Testing

- Manual testing is overhead for QA teams, error-prone

- Automated Unit Testing: Do this the most, because it narrows down which lines of code are the ones causing trouble

- Service Layer testing checks the application’s response to an input/output between the UI and the underlying unit code

- UI automated testing is messy and usually consumes more effort than the benefit extracted. Do this when it makes sense
How to Transition a Company to Scrum

When does it end?

- Transitioning to Scrum is a process of continuous improvement.
- There is no pre-defined end state so don’t talk about a complete transition.
- It is an iterative process where you try to get better and better.
Thank You!