Project Management at NetSuite

Jose Solorio
Senior Project Manager, NetSuite

February 12, 2015
1. Background
2. NetSuite
3. Project Management
4. Why Have a Methodology At All?
5. Waterfall
6. Scrum
7. How to Transition a Company to Scrum
Background

• Senior Project Manager & Scrum Master

• 7 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
Caster The App
• 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
Who We Are

- #1 Cloud Business Management Software
  - Cloud means hosted and maintained online
- 2400+ employees
- Used by 20,000+ organizations
- Used across 100+ countries
What We Do

NetSuite’s software allows businesses to operate, automate, and manage their business capabilities using a single application:

- **Financial Management**
- **Supply Chain Management**
- **Procurement**
- **Ecommerce**
- **Revenue**
- **HR**
- **Customer Service**
- **Partners**
- **Marketing Automation**
Locations

US
- San Mateo, CA (HQ)
- Paso Robles, CA
- Denver, CO
- Boston, MA
- Austin TX
- New York, NY
- Oklahoma City, OK
- Waterloo, Canada

Intl.
- Australia
- Canada
- Czech Republic
- Singapore
- United Kingdom
- Japan
- Philippines
- Hong Kong
- Uruguay
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**: Understanding of the work
  - **Tools**: Using software to measure progress
  - **Experience**: Knowing what to do in situations

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

- Methodologies are processes that teams follow that define what to do, the sequence to do it in, etc.
The Role of the Project Manager

- **Planning:** Working with team members to come up with a solid plan
- **Tracking:** Having a complete understanding of the status of the project at all times
- **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
Why Have a Methodology At All?
Let's 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. “
These are big decisions, so organizations need plans to figure out what direction to take, how money 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 to move forward.
Let's 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 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 either:
- Finish On-Time
- Early
- Late
- Or not even at all!
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?
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.*
Answer #2

Lets assume you went ahead and spent all this time making an elaborate 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!
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.
Waterfall
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><img src="image" alt="Timeline" /></td>
</tr>
<tr>
<td>Design</td>
<td><img src="image" alt="Timeline" /></td>
</tr>
<tr>
<td>Coding / Implementation</td>
<td><img src="image" alt="Timeline" /></td>
</tr>
<tr>
<td>Integration</td>
<td><img src="image" alt="Timeline" /></td>
</tr>
<tr>
<td>Testing and QA</td>
<td><img src="image" alt="Timeline" /></td>
</tr>
<tr>
<td>Deployment</td>
<td><img src="image" alt="Timeline" /></td>
</tr>
<tr>
<td>Maintenance</td>
<td><img src="image" alt="Timeline" /></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
Waterfall

But more often that not, things never go as planned.

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

...But if you are using Scrum, you do just enough planning.
Scrum
Scrum – Iterative Framework

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).

| Sprint X |
|---|---|---|---|---|---|---|---|---|
| **Week 1** | **Week 2** |
| Monday | Tuesday | Wednesday | Thursday | Friday | Monday | Tuesday | Wednesday | Thursday |
| Sprint Planning | Sprint Work | Sprint Work | Sprint Work | Sprint Work | Sprint Work | Sprint Work | Sprint Work | Sprint Work |
| | | | | | | | | | Sprint Retrospective |
| | | | | | | | | Update Product Backlog |

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

SW = Software

QA = Quality Assurance

** SM = Scrum Master

PO = Product Owner

Design Manager

QA Manager

Board of Directors

Diagram showing the roles and team structure in Scrum.
Prioritizes the team’s work

Take 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
Product Owners prioritize based on Risk-Value.

Product Owners prioritize based on Risk-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.
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 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)
## 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 WE accomplished”, never “look what I 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
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?
You’ve been called to manage a team that will create the next NetSuite Mobile App using a Scrum methodology.
<table>
<thead>
<tr>
<th>Week 1</th>
<th>Week 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monday</strong></td>
<td><strong>Monday</strong></td>
</tr>
<tr>
<td>Sprint Planning</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td><strong>Tuesday</strong></td>
<td><strong>Tuesday</strong></td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td><strong>Wednesday</strong></td>
<td><strong>Wednesday</strong></td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td><strong>Thursday</strong></td>
<td><strong>Thursday</strong></td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td><strong>Friday</strong></td>
<td><strong>Friday</strong></td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Sprint Work</td>
<td>Sprint Work</td>
</tr>
</tbody>
</table>

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

**Sprint X**: A weekly cycle of activities including Sprint Planning, Sprint Work, Sprint Review, Sprint Retrospective, and Update Product Backlog.
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 1.0

<table>
<thead>
<tr>
<th>Backlog</th>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to search</td>
<td>Ability to search</td>
<td>View Income</td>
<td>View Expenses</td>
</tr>
<tr>
<td>Reorder</td>
<td>Reorder</td>
<td>See saved searches</td>
<td>View list</td>
</tr>
<tr>
<td>Home button</td>
<td>View Sales</td>
<td>Home button</td>
<td>Customers</td>
</tr>
<tr>
<td>Calendar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leads</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>More options</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See saved searches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Sales</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Income</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Expenses</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View list</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have report snapshots</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View in Landscape Mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Highest Priority
(High-Risk, High-Reward)

### Lowest Priority
(Low-Risk, Low-Reward)
# Sprint Review, Sprint Retrospective, Update Product Backlog

## Sprint 1

### Sprint Planning

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Week 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>Sprint Review</td>
</tr>
<tr>
<td>Tuesday</td>
<td>Sprint Retrospective</td>
</tr>
<tr>
<td>Wednesday</td>
<td>Update Product Backlog</td>
</tr>
<tr>
<td>Thursday</td>
<td>Sprint Work</td>
</tr>
<tr>
<td>Friday</td>
<td>Sprint Work</td>
</tr>
</tbody>
</table>

### Sprint Work

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

### Sprint Review

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

### Sprint Retrospective

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

### Update Product Backlog

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Week 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>Update Product Backlog</td>
</tr>
<tr>
<td>Tuesday</td>
<td>Update Product Backlog</td>
</tr>
<tr>
<td>Wednesday</td>
<td>Update Product Backlog</td>
</tr>
<tr>
<td>Thursday</td>
<td>Update Product Backlog</td>
</tr>
<tr>
<td>Friday</td>
<td>Update Product Backlog</td>
</tr>
</tbody>
</table>
Sprint Review

Sprint 1

<table>
<thead>
<tr>
<th>Ability to search</th>
<th>✓</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorder</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>View Sales</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Sales: 446.6%
“OK, now that we all agree, let’s all go back to our desks and discuss why this won’t work.”
## Update Product Backlog

<table>
<thead>
<tr>
<th>Backlog</th>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calendar</td>
<td>Ability to search ✔️  Reorder ✔️  View Sales ✔️</td>
<td>View Income</td>
<td>View Expenses</td>
</tr>
<tr>
<td>Leads</td>
<td></td>
<td>See saved searches</td>
<td>View list</td>
</tr>
<tr>
<td>More options</td>
<td></td>
<td>Customers</td>
<td>Home button</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Customers</td>
</tr>
<tr>
<td>Have report snapshots</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View in Landscape Mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moved More Options from Backlog Sprint 3

Moved Customers from Sprint 3 to Sprint 2

**Highest Priority**
(High-Risk, High-Reward)

**Lowest Priority**
(Low-Risk, Low-Reward)
We start the cycle all over again.
Sprint 2

Fast-Forward
Sprint 2

- View Income
- See saved searches
- Customers

KPIS

Sales

446.6%

SAVED SEARCHES

- Customers by O/S Balance
- Purchase Price Variance By Cost Category
Sprint 3

- View Expenses
- View list
- Home button
- More options

Sprint 3

KPIS

<table>
<thead>
<tr>
<th>Sales</th>
<th>Income</th>
<th>Expenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>446.6%</td>
<td>$3,274,498</td>
<td>90.1%</td>
</tr>
</tbody>
</table>

Saved Searches

- Customers by O/S Balance
- Purchase Price Variance By Cost Category
Release 1.0

<table>
<thead>
<tr>
<th>Sprint X</th>
<th>Sprint X + 1</th>
<th>Sprint X + 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mobile Apps Theme
Finished Product!
How to Transition a Company to 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 and comes with 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 may have to be unlearned
• You will encounter resistance from people. WILL.
How to Transition a Company to Scrum

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 Execute An Adoption of Scrum, Follow 5 steps: ADAPT

- **Awareness** that current process is not working
- **Desire** to adopt Scrum to address existing issues
- **Ability** to succeed with Scrum
- **Promotion** of Scrum through sharing of other successful experiences
- **Transfer** the implications of using Scrum through the company
How to Transition a Company to Scrum

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.
How to Transition a Company to Scrum

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
How to Transition a Company to Scrum

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
How to Transition a Company to Scrum

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
How to Transition a Company to Scrum

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
How to Transition a Company to 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

Credits: Mike Cohn

Medium-size ✓
Medium-duration ✓
Medium-high importance ✓
Medium-high business sponsorship ✓
Pushing towards Agility

Once you select a project(s), a team(s), and an approach, 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
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 given 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
Technical practices to apply

Collective ownership. No one should say “That code belongs to John, 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

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
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