Implementing Agile methodologies with offshore personnel: *Acting fast from coast to coast.*

Yakir Shochat
Who are we?

Prion Network (formally known as IncrediMail) is a global consumer internet company that develops applications to make the online experience of its users simple, safe and enjoyable. Perion’s three main consumer brands are: IncrediMail, Smilebox and SweetIM.

2000 – Founded as ‘IncrediMail’
2006 – Gone public (NASDAQ ‘MAIL’ – now ‘PERI’)
2010 – Acquired USA based company ‘Smilebox’
2011 – Changed the name to ‘Perion’
2012 – Acquired Israeli internet company ‘SweetPacks’
2013 – Reported 2012 revenues increased to a record of $61.2 million

Perion Network is considered an Israeli internet pioneer and one of the leading Internet companies in the field with over 300 million downloads to date and more than 50 million monthly unique visitors across all of its brands.
Over the last decade the testing and development worlds have implemented and followed two separated processes to improve quality and minimize budget/timeline: Agile Methods and offshoring.
Introduction

Over the last decade the testing and development worlds have implemented and followed two separated processes to improve quality and minimize budget/timeline: Agile Methods and offshoring.

The Agile manifesto states: "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."

_Taken from [http://agilemanifesto.org/principles.html](http://agilemanifesto.org/principles.html)_

Offshoring definition according to Wikipedia: “Describes the relocation by a company of a business process from one country to another”.

_Taken from [http://en.wikipedia.org/wiki/Offshoring](http://en.wikipedia.org/wiki/Offshoring)_

Seems like both methods are contradicting each other.

Can we implement both successfully?
The story so far...

In October 2010, as one of the major organizational changes of the company, Perion Corp (then known as ‘IncrediMail’) has begun a long and complex course of outsourcing the QA department to TFT India.

The QA department consisted of 20 test engineer positions who have been outsourced.
The story so far...

In October 2010, as one of the major organizational changes of the company, Perion Corp (than known as ‘IncrediMail’) has begun a long and complex course of outsourcing the QA department to TFT India.

The QA department consisted of 20 test engineer positions who have been outsourced.
The primary driving force behind offshoring is cost. Organizations which want to reduce the development cost, started offshoring in countries which give them this opportunity. Beside this, some other reasons behind offshoring are: access to skilled labor, experience, time shifting, time to market, market access, extending venture capital money, and increased team flexibility and ability.
The primary driving force behind offshoring is cost.
Organizations which want to reduce the development cost, started offshoring in countries which give them this opportunity.
Beside this, some other reasons behind offshoring are: access to skilled labor, experience, time shifting, time to market, market access, extending venture capital money, and increased team flexibility and ability.

Although offshoring has advantages and benefits, there are few challenges faced by offshore development like:
• Geographical distance
• Time difference
• Cultural issues
• Linguistic barrier
• Poor communication and knowledge exchange/transfer
If you look at the numbers - the offshoring project was successful:
- Overall Testing budget was decreased in more than 50%
- Perion revenue per employ measurement increased (gross revenue/number of employees)
- Product quality remained in-tact

Numbers don’t lie…

But… What if the numbers don’t tell the entire story…?

What is the price for:
- Project delay and overhead
- Increased testing pipeline
- Decreased uploads per quarter
- Testing prestige status
- Lack of proximity to testing staff

Is there another side of this story?
The Plot Thickens

Two Sides To Every Story
After taking an in-depth look the entire picture is reveled:

- Project delay and overhead = INCREASED COSTS
- Increased testing pipeline = INCREASED COSTS
- Decreased uploads per quarter = DECREASED POTENTIAL REVENUES
- Testing prestige status harmed = DISRESPECT TOWARDS TEST AND DEVELOPMENT TEAM
- Lack of proximity to testing staff = LESS TESTS IN DEV

Suddenly the project ROI seems to be much lower...
The options

There were 3 considerable options

A. Accept the situation: comply to the new ‘rules’ current limitations
B. Return to in-house: rebuild the entire testing staff
C. TPI: Improving the test process

We have selected option C
But now we had to face a new question:
**Will improving just the test process would be sufficient?**

**In order to achieve real and measureable improvement - there is a need to redefine the entire process.**
In this case we require a comprehensive solution across the entire company.
The changes will impact the following:

- Product
- Project
- Marketing
- Development
- Testing
The Solution?
'Agile' is a nice 'buzzword' – but 'buzzwords' won't solve our problems
There are many types of Agile Methods.
Which one is suited more for us?
‘Agile’ is a nice ‘buzzword’ – but ‘buzzwords’ won’t solve our problems. There are many types of Agile Methods. Which one is suited more for us?
Globalization & Agile together seems to contradict each other.

The Agile Manifesto was written in 2001, before global development\test teams where so popular.

Principles behind the Agile Manifesto

We follow these principles:

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

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

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

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

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

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

Working software is the primary measure of progress.
In offshore development\Testing, distributed team members are far apart from each other. There are many challenges in managing distributed team members, mostly poor communication, control and complex knowledge exchange.

Scrum is an agile method that demands a close collaboration between developers, customer/product owner and the project manager.

Agile in general promotes just enough documentation while offshore best practice is formalized and detailed documentation.

This collaboration suffers when project members are far apart from each one another.
Issues: Scrum and offshore

- Communication and collaboration
- Knowledge transfer
- Lack of project transparency
- Project tracking
- Handling Ad-hoc items
- Risks: Communicating and handling risks
- Participating in Scrum ceremonies
- Lack of trust within the teams
## Challenges

<table>
<thead>
<tr>
<th>Factors</th>
<th>Challenges of offshore development</th>
<th>Prerequisites for implementing Scrum</th>
</tr>
</thead>
</table>
| Geographical distance | Invisible development process  
Problems in communication and knowledge exchange  
Limited face to face meeting                                                                                                       | Co-located, self-organized team  
Visible process, Frequent iteration  
Required everyone to communicate and share knowledge  
Scrum ceremonies focus on daily face to face meetings                                                                                      |
| Time Zone differences | Asynchronous communication  
Problems solving hard and slow  
Difficult in monitoring the work                                                                                                       | Synchronous communication  
Fast feedback, quick response to customers requirements change  
Scrum require symmetric work schedule                                                                                                      |
| Cultural differences | Difficult to develop trust and relationship between team.  
Misunderstanding increase because lack of awareness, interpretation and meaning of different terms.                                    | Scrum focus on team building, trust and relationship  
Demand awareness, visibility of process and transparent project documentation                                                               |
| Linguistic barrier  | Unable to communicate with remote site, results in low communication.  
Misunderstood, unclear and ambiguities in project documentation                                                                                   | Required everyone to Frequently communicate and share knowledge  
Transparent project documentation                                                                                                             |
| Documentation       | Require in deep in depth documentation for full redundancy and to compensate issues in knowledge transfer.                                           | Lean documentation                                                                                         |
People are different.
We have different look, food, different cloths and different beliefs... That is what makes this world great!

We also have different culture, which impacts are working habits - for good and for bad

The average Indian engineer are highly skilled and diligent – but also almost never comfortable saying “No” to anyone...

Not even to their direct manager, project manager or any other major stakeholder.

This can cause issues related to task estimations, project schedule and project transparency.
Process?

Processes are like haircuts...
Copying someone else’s rarely works
One size doesn't fit all

Implementing a process ‘by the book’ seems to be little awkward according to the current situation. Sometimes to achieve optimal results, processes and methodologies needs to be tailored according to the origination needs, culture, infrastructure, product and regulations.
The Answer?

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

**Individuals and interactions** over processes and tools
**Working software** over comprehensive documentation
**Customer collaboration** over contract negotiation
**Responding to change** over following a plan

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

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas
Building The Foundations

Agile is built on trust - meeting face-to-face is the only way to build it!

- Budget-in recurring face-to-face meetings between local team, and offshore team
- Having strong communication infrastructure like Skype, Video Conference, Phones, high bandwidth, big screens to facilitate virtual team etc.
- Seed visits in frequent intervals across the length of the project: both offshore and onshore teams sitting, working and partying together is a must to ignite relations and build strong trust
- The highest performing teams are those based on trust
Agile is built on trust - meeting face-to-face is the only way to build it!

- Budget-in recurring face-to-face meetings between local team, and offshore team
- Having strong communication infrastructure like Skype, Video Conference, Phones, high bandwidth, big screens to facilitate virtual team etc.
- Seed visits in frequent intervals across the length of the project: both offshore and onshore teams sitting, working and partying together is a must to ignite relations and build strong trust
- The highest performing teams are those based on trust
Building The Foundations

Agile is built on trust - meeting face-to-face is the only way to build it!

- Keep in mind that the ‘exchange program’ can be logistically problematic and not cost effective – the right ratio and occurrence must be planned according to the project needs
- Training session have been proven to be very effective when done on-site

Establish a shared project vision

- Participation in this activity by whole team emphasizes ownership of the project results
- Use collocation travel as best opportunity to build this
- Doing release planning, sprint planning and daily scrum together with offshore team builds similar shared vision

Involve full team in release planning, iteration planning, review, and retrospectives

Everyone on the team participates in the sprint planning meetings, sprint reviews and retrospectives
If sub-team do their own retrospective, then share results with others teams
Building The Foundations

**Agile is built on trust – building a relationship**

- Avoid teams to be categorized according to their culture, location etc. – no more categorization about ‘The Indians” or “The Israelis” - its Development and Testing teams
- No more ‘We’ and ‘Them’ – There is ‘US’! We are one unified group working for a mutual goal
- Treat your offshore teams as if they were your friends, your equal, and nothing less than company employs – you will achieve much more with this approach!
- Encourage proximity to the other teams, the company under a mutual goal. Think outside the box!
Agile is built on trust – building a relationship

- Avoid teams to be categorized according to their culture, location etc. – no more categorization about ‘The Indians” or “The Israelis” - its Development and Testing teams
- No more ‘We’ and ‘Them’ – There is ‘US’! We are one unified group working for a mutual goal
- Treat your offshore teams as if they were your friends, your equal, and nothing less than company employs – you will achieve much more with this approach!
- Encourage proximity to the other teams, the company under a mutual goal. Think outside the box!
Some positions are better off not to be distributed - but preferred to remain In-house.

Always have at least one focal point that communicates with the offshore team in house.
The average ration of 1 representative per 5-6 offshore personal have been found to be successful.
It is advised that personal will act as ‘Team Leader’ and if possible also as Scrum Master or coordinator between the teams.
Improving The Process

Use short sprints
- Offshore creates poor visibility and control than in-house personal
- Short iterations are much easier to control, review and gain knowledge of process improvement and it’s implementation over time
- Short iterations insure that you see what the offshore team is doing on a regular basis.
- Allows quick feedback and redirect the team quickly
- We have selected a 2 weeks sprint time period
- Add to spring estimation the knowledge transfer time period according to complexity

Have PM/TL to execute at least 1 sprint with entire team at offshore site
- Build and improve relations
- Get familiar with the working environment in offshore site – and look for points to improve
- New issues and difficulties will be uncovered – along with new solutions
- Offshore team will get to build relation directly with project manager/team leader and get to hear requirements directly
- Agile/Scrum ceremonies are now take place in offshore site as the main host
- Repeat once in every visit – and always try to improve the process
ScrumMaster at both locations
- Scrum Training officer in both in-house and offshore site
- One focal point at each site – for better communication

Data sync on both sites
- Requirements, documentation, code update should be synchronize immediately
- ‘Site To Site VPN’ to allow offices in multiple fixed locations to establish secure connections with each other
- Site-to-site VPN extends the company's network, making computer resources from one location available to employees at other locations
- Use web based ALM that support Agile methodologies
- Shared folders should be available to all stakeholders on all sites – don’t use replications
Establish a synchronization and communication plan

• Communication management is the key element in offshore project planning and coordination
• Define how local team and offshore team will communicate and maintain synchronization
• Inspect, select and define which communication tool is right for which propose – PROS VS. CONS
• The tools selection and definition might require interviews and pilot sessions to take place
• Define daily and distributed stand-ups, retrospectives and sprint review time – not one on each site – but one unified Agile ceremonies schedule
• Dedicated resource to communication
Email
Communication through email is useful but also has it’s limitations:
• Though Email is a fast communication channel but sometimes the reply is received late from the other end
• One unified languages is a must – mostly when multiple recipients are involved

Instant messenger/Skype
• Very accessible – but most be used for the right reasons
• Less formal than email
• Data is not indexed – difficult to search
• ‘Presence’ feature not always effective
• Not formal and documented
• Should be used as on-going communication tool
Video conference

- A video conference is an excellent tool to provide face to face communication
- Match names to faces
- Ideal for Agile/Scrum ceremonies: Sprint planning, daily meeting and sprint review
- The offshore team are a part of the planning, estimation and review – which strengthen their bond to the project, help them better understand the rational behind specific concepts and improves overall testing
- The most cost effective solution today
Remote connection

- Great tool to reproduce defects in real-time on off-site configuration
- Can be used to perform acceptance tests among many stakeholders conference
- Debugging in real time is also possible when using remote debugger and debug tools at testing end
- No tool is required for joint server domain
- Many tools available for external server domain

Defect tracking tool/ALM

- Should be web based and not client based for online updates
- Support Agile process (Sprints schedule, Scrum/Kanban board, stories tasks etc)
- Cab accessible from any machine, in our out of the domain (can bare a security issue)
- Updated in real time
- Should have chat/email integration for better communication
Can we turn the tables?

Working schedule – the hidden risk!

In Israel the formal (and optimal) work day in the IT industry is 9 hours. Usually from 09:00-10:00 until 18:00-19:00. The same working day schedule takes place in India.

So far so good, but what about the time zone difference?

The time zone difference 2.5 hours in Israel’s ‘winter time’ and 3.5 hours during ‘summer time’! Average of 3 hours per day over a yearly schedule.
Can we turn the tables?

Working schedule – the hidden risk!
In Israel the formal (and optimal) work day in the IT industry is 9 hours. Usually from 09:00-10:00 until 18:00-19:00. The same working day schedule takes place in India.

So for so good, but what about the time zone difference?
The time zone difference 2.5 hours in Israel’s ‘winter time’ and 3.5 hours during ‘summer time’! Average of 3 hours per day over a yearly schedule.

<table>
<thead>
<tr>
<th>Time (Israel)</th>
<th>Project in-house team member (R&amp;D)</th>
<th>Project offshore team member (testing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6:00 - 7:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:00 - 8:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:00 - 9:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:00 - 10:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00 - 11:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:00 - 12:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:00 - 13:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13:00 - 14:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:00 - 15:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00 - 16:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:00 - 17:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17:00 - 18:00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Only 6 hours corresponding
Can we turn the tables?

Working schedule – the hidden risk!

In Agile we require quick feedback, smooth communication flow between teams and synched participations in specific process ‘ceremonies’ (daily, sprint planning etc.).

Since in Agile the working schedule is divided into sprints, a crucial question arise: Does a sprint capacity in Offshore site is the same as a sprint performed ‘In-house’?

Like in every distributed project, there are time differences between one site to the other. In Israel we work from Sunday to Thursday while in India they work from Monday to Friday.

On paper – a 14 days sprint will have 10 testing days on either if this team is distributed or in house, as the ‘missing’ day in the schedule is ‘moved’ to Friday.

In other words: Friday can ‘compensate’ on Sunday. Is that so?

| In-house          |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
|-------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Team              | Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | total hours |
| R&D team (total of 5 members) | 45     | 45     | 45     | 45    | 45    | 45     | 45     | 45     | 45     | 45     | 45     | 45     | 45     | 45     | 450 |
| Test team (total of 4 members) | 36     | 36     | 36     | 36     | 36    | 36     | 36     | 36     | 36     | 36     | 36     | 36     | 36     | 36     | 360 |

| Offshore          |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
|-------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Team              | Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | total hours |
| R&D team (total of 5 members) | 45     | 45     | 45     | 45    | 45    | 45     | 45     | 45     | 45     | 45     | 45     | 45     | 45     | 45     | 450 |
| Test team (total of 4 members) | 36     | 36     | 36     | 36     | 36    | 36     | 36     | 36     | 36     | 36     | 36     | 36     | 36     | 36     | 360 |
Can we turn the tables?

Working schedule – the hidden risk!

In Israel we work from Sunday to Thursday while in India they work from Monday to Friday.

On paper – a 14 days sprint will have 10 testing days on either if this team is distributed or in house, as the ‘missing’ day in the schedule is ‘moved’ to Friday.
In other words: Friday can ‘compensate’ on Sunday.
Is that so?

<table>
<thead>
<tr>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In-house</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td></td>
<td></td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td>Test team (total of 4 members)</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td></td>
<td></td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>360</td>
</tr>
<tr>
<td><strong>Offshore</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td></td>
<td></td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td>Test team (total of 4 members)</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td></td>
<td></td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>360</td>
</tr>
</tbody>
</table>

In vivo - the result is much different.
On Sunday - the project lacks in testing feedback on daily build/version – which causes a daily in the entire project
On Friday - the testing process is not fully utilized because of lack for R&D feedback/respond – which causes further delay!

We are on the losing ends at both sides!
Can we turn the tables?
Working schedule – the hidden risk!

Keep in mind that lack of in-house supervision\feedback – is a major risk!

Example day to day scenarios that can be solved within minuets at in-house team can take days in offshore team:

• A build/version that was released on Thursday evening is only tested the next day – and showstopper issue that has been identified and reported on Friday will only be recognized and rectified on Sunday, and re-tested again on Monday!

CAUSES A DELAY OF 3 DAYS IN THE PROJECT SCHEDULE.

Murphy's Law...
What if the defect reporting is ambiguity/not clear/incomplete/missing?
What if the there is an issue with the requirement/release notes?
For both issues – a the missing information will only be received the on Monday!

CAUSES A DELAY OF 4 DAYS IN THE PROJECT SCHEDULE
Can we turn the tables?

Agile is not just about process oriented procedure – it is also a **mentality**!

- Creative and Flexible stage of mind
- Think ‘outside the box’
- Versatile and capable of improvising
In Israel, for religious reasons, the weekly working schedule is between Sunday to Monday. In India there is no religious law that dictate the weekly working schedule.

Switching the working schedule might cause some discomfort to a few but perfectly legal. Remember: almost any issue can be solved with the right incentive.

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>Test team (total of 4 members)</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>360</td>
</tr>
</tbody>
</table>

We have decided to initiate a pilot in one of the testing teams, and to align the testing weekly schedule with the rest of the project.

<table>
<thead>
<tr>
<th>Phase 1</th>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>Test team (total of 4 members)</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>360</td>
</tr>
</tbody>
</table>

We have measured the test team productivity, feedback, capacity and entire project schedule for 2 sprints. Feedback from other stakeholders within the project have been collected as well. The results have indicated that the project capacity and productivity have been increased while resulted with minimum suspension criteria and blocking schedule reached.

How can we make the best of it? How can we improve? After aligning the testing schedule from Sunday to Thursday we have initiated phase 2: splitting 50% of the testing team on Sunday and reallocating them on Friday.

<table>
<thead>
<tr>
<th>Phase 2</th>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>Test team (total of 4 members)</td>
<td>18</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>18</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>360</td>
</tr>
</tbody>
</table>

By doing so, we will gain a testing feedback across 6 days a week! A pilot have been initiated and have proven to be a success – the testing process have been improved, feedback have been received more rapidly and for the first time at both sprints ends, no backlog tasks or technical debt was required.
Yes we can!

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>Test team (total of 4 members)</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>360</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase1</th>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>Test team (total of 4 members)</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>360</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase2</th>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>Test team (total of 4 members)</td>
<td>18</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>18</td>
<td>360</td>
</tr>
</tbody>
</table>

We have improved the testing process which resulted in an improvement across the entire project, however - the risk of lack of in-house supervision/feedback on Friday remained.

To mitigate the risk by having in house personal available on Friday would require management to be extremely careful.

It was decided that a routine between In house Test Team\Group Leader and Project manager will log to their work station Twice on Friday.

<table>
<thead>
<tr>
<th>Phase3</th>
<th>Team</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>total hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R&amp;D team (total of 5 members)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>Ongoing</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>Test team (total of 4 members)</td>
<td>18</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>18</td>
<td>360</td>
</tr>
</tbody>
</table>

**First** – in mid day to check if all is working well in offshore site, to issue feedback, solve issues and activate other alternatives if required.

**Second** – To read daily status at the end of the day

There were no scheduled working hours and they were logged according to personal trust within the members.

With today’s technology and in standard ‘no crisis’ normal routine, those tasks could be done from any smart phone - anywhere.

**After measuring the working hours per 6 sprints – each session averaged less than 30 minuets.**
The Tables Have Been Turned!
Thank you!