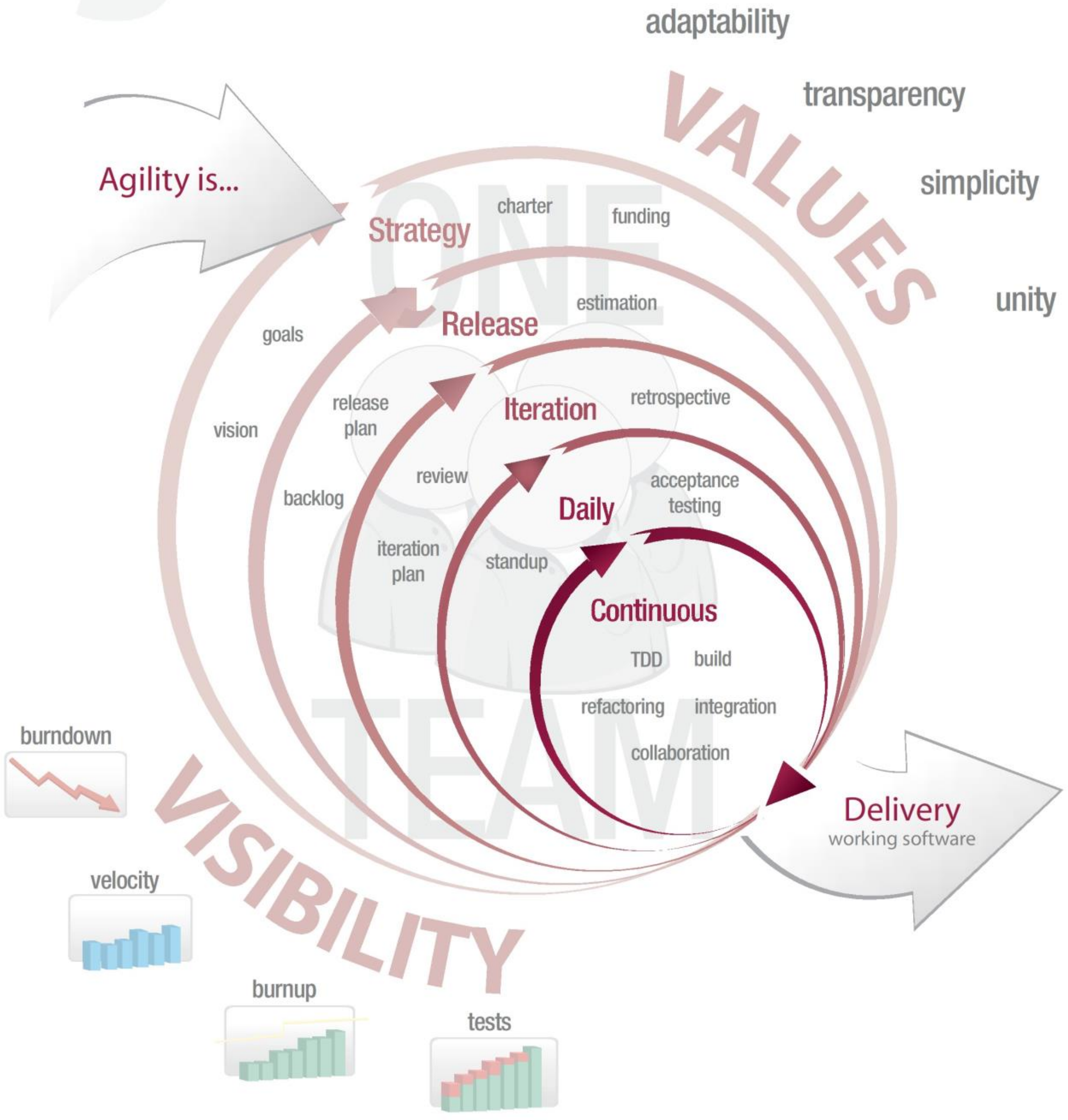
Agile Development



Accelerate Success

Author & Instructor: Vijay Nathani

Assignments

Exercise: Existing Processes
How effective are the existing processes and development practices within your organization?

10 == couldn’t be better; living the dream!
1 == failing on multiple levels; train wreck waiting to happen

Exercise: Defined vs. Empirical Processes
Part 1

1. Form pairs.
2. Assign one person the boss, the other is the worker.
3. The boss can give the following commands . Go, Stop, Right, Left, Faster, Slower
4. The worker must follow the boss’ commands.
5. The boss is responsible for having the worker proceed 60 normal paces to a common destination within two minutes, from the time “go” is said, until “stop” is said, by the moderator.
6. The boss can command the worker but not touch the worker.

Part 2

1. With the same teams as before, except everyone is a worker and responsible for figuring out how to proceed during the exercise by him or herself.
2. Each team proceeds 60 normal paces to a common destination within two minutes, from the time “go” is said, until “stop” is said, by the moderator.

Exercise: MLBTix

Overall attendance at baseball games has increased over the last 10 years. In some cities, such as Boston, almost all games are sold out, and obtaining tickets through normal channels is nearly impossible. MLB rules prohibit the resale of tickets at a profit. Scalping (resell something for a quick profit) is illegal and has been cracked down on recently. The primary distribution channel for buying tickets is an online auction site, xAuction. Although all auctions for tickets on xAuction are supposed to be capped at the retail price plus expenses, MLB has learned that, through a variety of workarounds, these tickets are being scalped for prices of up to 1000 percent of the retail price.

Project Plan

The MLB commissioner’s office hired an external consulting organization, Denture, to plan a project to manage the resale of baseball tickets. Denture delivered the final plan on November 15, and it was subsequently approved. Excerpts of the plan are provided here.

Project Background New legislation mandates that as of the 2011 baseball season all ticket resales must take place through MLB-authorized facilities. MLB has decided to develop such a facility on the Web; the site will be known as MLBTix. Through functionality similar to the online auction site, xAuction, but specific to MLB, the public will be able to buy and sell

MLB tickets online. Sellers will auction the tickets to the highest bidder, setting an initial bidding price of their own choice without floor or ceiling conditions established by MLBTix. The seller can also limit the duration of the auction by setting a start and an end date and time. If the ticket(s) are successfully sold, the buyer pays the seller through the MLBTix credit card facilities, and the seller mails the tickets to the buyer. Sellers will automatically be notified when buyers receive their tickets, at which point MLBTix will mail a check for the proceeds (less the 25 percent MLB fee) to the seller.

The commissioner will be announcing MLBTix at a news conference on January 15. He hopes to have some functionality available by opening day, March 30, 2011, and for the site to be fully functional by the All Star break, which begins July 18, 2011. Therefore, March 30, 2011 is the anticipated release date. On this date, the MLBTix site will be up, and buyers and sellers will be able to register. Sellers will be able to make tickets available at a fixed price, which buyers will be able to pay in full via credit card. MLBTix is a go-between, but all tickets are transferred directly from sellers to buyers. The release schedule mandates that on June 30, 2011, auction capability be added to site. Finally, on August 30, 2011, buyers will be able to buy groups of collocated tickets, view the locations of seats being sold, and check on inventory.

Funds for the project are ample and should not be considered an unreasonable constraint. The deliverables are the date and functionality. Facilities or packaged software to support MLBTix can be either bought or developed— whichever helps meet the date. The commissioner needs a heads-up on the likelihood that the MLBTix will be available by the above dates prior to his press conference.

Product Backlog These are the functional requirements:

- Customers can register as potential sellers of tickets and be assigned a user ID and password.
- Customers can register as potential buyers of tickets and be assigned a user ID and password.
- Customers can maintain a profile under the user ID, including e-mail address, street address, preferences, and credit card information.
- Customers can place tickets up for auction, declaring a floor price, start of auction time/date, and end of auction time/date. Sufficient information should be provided so that buyers can ascertain that the tickets meet their requirements (for the right days, right teams, right number of seats located next to each other, and the seat locations in the ball park).
- Customers can cause an auction to be conducted for the tickets to registered buyers.
- Customer can have MLBTix successfully conclude the auction by awarding the tickets to the highest bidder by the end date and, at the same time, debiting the buyer’s credit card and placing the funds in an MLBTix account.
- MLBTix will notify the seller of the successful sale of the tickets and provide the delivery information for the buyer.
- MLBTix will provide the buyer with a mechanism for indicating that the tickets were not successfully received by the selling date plus a specified period of time (for example, one week).
- MLBTix will transfer the funds for the ticket sale less 25 percent to the seller at the end of the specified delivery time, unless the buyer has indicated otherwise.

- MLBTix will transfer the 25 percent plus any interest to a corporate MLB account from the MLBTix account automatically.
- MLBTix will provide customers with inventory and inventory search capabilities for teams, tickets, dates, and seats.
- MLBTix will provide for promotions on MLBTix.
- MLBTix will be able to identify and ban abusers of MLBTix.

These are nonfunctional requirements. MLBTix must be able to

- Handle 250,000 simultaneous users with sub-second response time.
- Be secure at the anticipated level of financial activity (2,000 tickets per day at an average selling price of $50).
- Be scalable to 1,000,000 simultaneous users if necessary.
- Be 99 percent available, 24 hours a day, 7 days a week.

This is the development context for bidders: The system will be created in a development environment for building Open Source products, using Intel technology and software running on an OpenSQL database server. The development Team members will all live within easy commuting distance of the development site. There are currently cubicles at the development site. The development environment is wireless and has all power and networking capabilities already operating. The development environment uses Open Source development tools such as Eclipse. The development Team is required to use a source code library, check in code every time it’s changed, build the software at least daily, and unit and acceptance test the software every time that it is built. Scrum will be used as the development practice. Use of any other aspects of Extreme Programming or any other engineering practices, such as coding standards, is up to the Team. All of the developers on the Team must have excellent engineering skills and at least be familiar with Scrum and Extreme Programming. The Team must consist of development engineers with excellent design and coding skills. These engineers are responsible for all testing and user documentation, although they can hire contractors to assist with this. The engineers on the Team must average 10 years of progressive experience on software projects using complex technology and Open Source software products. All Team members must be baseball aficionados.

The Project

Imagine that after a quick request for proposal (RFP) process, the commissioner of MLB selected your organization to develop MLBTix. In your response to the RFP, you assured the commissioner that you can meet the release schedule. You were present with the commissioner at a press conference on January 15 when he announced MLBTix, and at this press conference, you demonstrated the functionality completed during the first Sprint. This Sprint began on December 7, 2010, and the Sprint review meeting was held on January 7, 2011.

Your Team has just completed its third Sprint, which ended on March 7, 2011 (assume today is March 8, 2011). You have demonstrated the functionality developed during this Sprint to the commissioner. All the functionality necessary for the first release is in place. You intend to pull everything together into the production environment for the planned initiation of MLBTix on March 30, 2011, the start of the MLB 2011 season.

Uh-Oh!

At the Sprint planning meeting for the fourth Sprint, you and the Team become concerned about the capability of MLBTix to handle the kind of volumes that might be encountered. MLB has hired a public relations firm to market MLBTix, and it's done almost too good of a job: MLBTix has been the rage of every sports page and sports magazine. Everyone who knows about baseball knows about MLBTix and knows that it will be available as of 12:00 p.m. Eastern time on March 30, 2004. There are over 40 million baseball fans, and you know that almost no system could handle 40 million simultaneous hits.

You provide the commissioner with the following background information: The Team contacted several e-commerce retailers and determined that there would be on average 100 visits for every sale. The Team is unable to estimate the exact number of hits that will occur when the Web site first goes up but is worried that it will be more than it can handle. The MLB commissioner's research indicates that the site will likely sell 2,000 tickets per day in April 2004 and 5,000 per day thereafter for the rest of the season. The average price that will be charged by a seller above retail is $30, of which 25 percent will go to MLBTix. You have previously alerted the commissioner that the database technology Denture recommended the Team use is an iffy proposition at best, and scaling tests have shown the application to be database intensive. Even with all the tuning efforts from the consultants that have been brought in, and even running OpenSQL on the fastest RAID devices possible, the maximum number of simultaneous transactions that can be served with sub-three-second response time is 100 per second. Loads are expected to reach significant peaks at lunchtime and after dinner. The Team is concerned that peak volumes during normal production might overwhelm the server and that the knee of the performance curve will be very close to the 110-transactions-per-second rate. You have determined that the Miracle database will readily support the scaling requirements predicted by the commissioner, but it will take one more Sprint to trade out OpenSQL and implement Miracle database. The upshot? The application won't be ready until a month after the season opener.

What Advice Should You Provide?

You tell the commissioner all of this. You notice that the commissioner gets increasingly agitated during your presentation, tapping his feet, spitting at the floor, and uttering muffled expletives. He appears to be very unhappy. The commissioner tells you to knock off all of this technology mumbo-jumbo and tell him what he should do. He wants to know whether he should call in his public relations people and tell them to announce that he can't get MLBTix up. What should you advise the commissioner based on the above risk/reward model and your best instincts?

Exercise – Complete Teams

1. Do you have operations people on your team?
2. Is someone from tech support regularly consulted when designing new features?
3. Are testers involved in development right from the start?
4. When do technical writers get involved?
5. Are customers or customer proxies treated as full team members?

Exercise – Processing in Large Batches vs. Small Batches

Let 5 people sit in a circle. Give one person a box of coins.
Round 1: Each person flips ALL coins. When done with the entire batch, gives all the coins to the next person.
Round 2: Each person flips one person and passes the flipped coin to the next person. Keep flipping and passing until done.

Exercise – Deming

Deming's System of Profound Knowledge says:

a. It's the whole product, the whole team, the whole system that matters.
b. When something goes wrong, in all probability it was caused by the system, which makes it a management problem.
c. Use the scientific method to change and improve.
d. With people, the things that matter are skill, pride, expertise, confidence, and cooperation.

Imagine that Deming was scheduled to tour your organization next week. Prepare a presentation for him on how each of these points is addressed in your organization. What do you think his advice would be? Would you be prepared to act on it?

Exercise – Fourteen Points

Review Deming's 14 points with your team. For each point think:

- Is this point relevant today in our organization? Is it important?
- If we regard it as relevant and important, what does it suggest we should do differently? What would it take to make the change?

The 14 points are mentioned below:

1. Provide for the long-range needs of the company; don't focus on short term profitability.
2. The world has changed, and managers need to adopt a new way of thinking. Delays, mistakes, defective workmanship, and poor service are longer acceptable.
3. Quit depending on inspection to find defects, and start building quality into products while they are being built. Use statistical process control.
4. Don't choose suppliers on the basis of low bids alone. Minimize total cost by establishing long-term relationships with suppliers that are based on loyalty and trust.
5. Work continually to improve the system of production and service. Improvement is not a one-time effort; every activity in the system must be continually improved to reduce waste and improve quality.
6. Institute training. Managers should know how to do the job they supervise and be able to train workers. Managers also need training to understand the system of production.

7. Institute leadership. The job of managers is to help people do a better job and remove barriers in the system that keep them from doing their job with pride. The greatest waste in America is failure to use the abilities of people.
8. Drive out fear. People need to feel secure in order to do their job well. There should never be a conflict between doing what is best for the company and meeting the expectations of a person's immediate job.
9. Break down barriers between departments. Create cross-functional teams so everyone can understand each-other's perspective. Do not undermine team cooperation by rewarding individual performance.
10. Stop using slogans, exhortations, and targets. It is the system, not the workers, that creates defects and lowers productivity. Exhortations don't change the system; that is management's responsibility.
11. Eliminate numerical quotas for workers and numerical goals for people in management. (We add: Eliminate arbitrary deadlines for development teams.) This is management by fear. Try leadership.
12. Eliminate barriers that rob the people of their right to pride of workmanship. Stop treating hourly workers like a commodity. Eliminate annual performance ratings for salaried workers.
13. Encourage education and self-improvement for everyone. An educated workforce and management is the key to the future.
14. Take action to accomplish the transformation. A top management team must lead the effort with action, not just support.

Exercise – Visual Workspace

In column 2, answer the question posed in column 1 as it relates to your organization.
In column 3, rate how self-directing your organization is.
Give your organization a score of 0 to 5, with

0 = people are told what to do and
5 = people figure out among themselves what to work on next.

Question	**Current Practice**	**Score**
1) How do people know what customers really want?		
2) How do people get technical questions answered?		
3) How do people know what features to work on next?		
4 How do people know what defects to work on next?		
5) How do people know if tests are passing?		
6) How do people know their progress toward meeting the overall goal of their work?		

Exercise – Employee Survey

Do you have an employee survey? Does your employee survey have these questions?

a. Do you feel that the compensation system is fair?
b. Do you feel that the promotion system is fair?
c. Rate how the compensation affects your dedication to your job (Scale of 1 to 5):
   1. It angers me and gets in the way of doing a good job.
   2. It sometimes annoys me and occasionally affects my performance.
   3. It doesn't make much difference.
   4. It occasionally motivates me to work harder.
   5. It motivates me to work hard every day.

How would it impact your organization if these questions are added?

Exercise – Problem

What, exactly, is our biggest problem? And what, exactly, are we going to do about it?

Exercise - Refactoring

1. What is your team's practice regarding refactoring?
2. Before you add any new features, do you first change the design to simplify it, without making any feature changes, and test the new design to be sure nothing has changed?
3. Do you refactor to simplify immediately after getting a new feature working?

Exercise – Cycle Time

As per Little Law:

$$\text{Cycle time} = \frac{Number\ of\ items\ in\ the\ process\ (work\ in\ progress)}{Throughput\ (How\ many\ items\ are\ produced\ per\ unit\ of\ time)}$$

What is the cycle time?

1. A team is producing production grade code every sprint. Sprint length is 4 weeks. Sprint backlog, on an average, is 50 user stories.
2. In the above example, what if the sprint length was 2 weeks (instead of 4) and average user stories done every sprint were 20 (instead of 50).
3. A team is producing integration tested code every 2 weeks sprint. They do a hardening cycle of 4 weeks after typically 10 sprints and then produce production grade code.

Exercise – Testing

On a scale of 0-5 (with 0 = low and 5 = high) rate your organization on:

a. Standardized architecture
b. Standardized tools
c. Coding conventions
d. Configuration management
e. Automated unit tests
f. Automated acceptance tests
g. One-click build and test
h. Continuous integration
i. Automated release
j. Automated installation

For any score that is below 3, is there room for improvement?
What stops your team from doing this improvement?

Exercise – The 5S

Ask the team to take a look at your team room. Rate its general appearance on a scale of 0-5 (high). Now rate the general neatness and simplicity of your code base on the same scale. Are the results similar? If you have a score of 3 or lower, propose to the team that you do a 5S exercise, first on the room and then on the code base.

The 5S for Java

1. Sort: Reduce the size of the code base. Throw away all unneeded items immediately. Remove:
    - Dead code
    - Unused imports
    - Unused variables
    - Unused methods
    - Unused classes
    - Refactor redundant code
2. Systematize: Organize the projects and packages. Have a place for everything and everything in its place.
    - Resolve package dependency cycles
    - Minimize dependencies
3. Shine: Clean up. Problems are more visible when everything is neat and clean.
    - Resolve unit test failures and errors (passed == 100%)
    - Improve unit test coverage (> 80%)
    - Improve unit test performance
    - Check AllTests performance
    - Resolve checkstyle warnings
    - Resolve PMD warnings
    - Resolve javadoc warnings
    - Resolve TODO's

4. Standardize: Once you get to a clean state, keep it that way. Reduce complexity over time to improve ease of maintenance.
5. Sustain: Use and follow standard procedures.

Exercise – Standards

Who is responsible for setting standards? Who should be? How closely are standards followed? How easily are they changed? Is there a connection between the last two answers in your organization?

Exercise – Contracting

What is the primary purpose of outsourcing in your company? What kinds of activities does the company keep inside rather than outsource? Are the incentives of outsourcing agreements aligned with the best interests of your company and your customers? Can you imagine ways in which your outsourcing arrangements might create questions of allegiance in the minds of workers in your company? In the other company?

For contracting companies: What kinds of contracts do you routinely use? What do you see as the key benefit of that kind of contract? What is the key risk? If you were in the shoes of your contractors, what would you see as the benefits and risks of that kind of contract?

For contractors: What kinds of contracts do you routinely engage in? Are some types better for you than others? Consider your favorite contracting format and reconsider it from the point of view of the contracting companies. What do you see as the benefits and risks for them?

Exercise – Scrum Product Backlog item

Product backlog item	**Estimate**
Read a high level, 10-page overview of agile software development in a business magazine.	
Read a densely written 5-page research paper about agile software development in an academic journal	
Write the product backlog for a simple eCommerce site that sells only clocks	
Recruit, interview and hire a new member for your team	
Create a 60-minute presentation about agile estimating and planning for your coworkers	
Read a 150-page book on agile software development	
Write an 8-page summary on this session for your boss	

In the above table, mark any item as 5 units in estimate. Give relative estimate for other items. You can choose only one of the following units: 1, 2, 3, 5, 8, 13, 20, 40, and 100.

If a task is above 40, then try breaking it into smaller tasks. Use the poker card method for estimate. Total time for this exercise is 10 minutes for a group of about 5 people.

Exercise – Scrum – Available time exercise

This exercise is to be performed by a group of 6 persons. If there are lesser people in any group, one person can perform two roles.

Aishwarya	Out of office on vacation the 12th and 13th. 5 total productive hours each day, but average 1 hour of maintenance responsibilities each day.
Abhishek	In training all day on the 20th. 6 total productive hours each day, and no maintenance responsibilities, but divides time 50/50 between this team and another
Shah rukh	No days out of the office planned. 6 total productive hours each day, but average of 2 hours of bug-fixing responsibilities each day.
Amitabh	Out of vacation the week of the 11th. 7 total productive hours each day, but average of 2 hours of maintenance responsibilities each day.
Hrithik	Out of office on the 13th. 5 total productive hours each day, and no maintenance responsibilities
Preeti	No days out of office planned. 6 total productive hours each day, but averages 1 hour of operational responsibilities each day.

The 4 week sprint is marked in the calendar below.

Mon	Tue	Wed	Thu	Fri
				1
4 Sprint Planning Meeting	5	6	7	8
11	12	13	14	15
18 Office Closed. (state holiday)	19	20	21	22
25	26	27	28	29 Sprint Review and Retro

Calculate the available working hours during the Sprint for the team.
Sprint Length – 4 weeks
Available working days during the Sprint – 17

Team Member	Available days during Sprint (Net of holidays & planned days out of office)	Available hours per day for Sprint	Total available hours during Sprint
Aishwarya			
Abhishek			
Shah rukh			
Amitabh			
Hrithik			
Preeti			

Exercise – Scrum Release Planning Exercise

You are a team that's planning a release. Start by calculating the team's velocity, based on the last 3 sprints. Then look at the product backlog and answer the following questions

- How much of the product backlog could you commit to release by May 15? (Date-driven release)
- On what date could you commit to a release with product backlog items 1-15? (Feature-driven release)
- Could you commit to a release with product backlog items 1-32 by June 1? (both date-driven and feature-driven)

Data:

- Sprint begins on 1-Jan-2011
- All sprints are 4 weeks in length
- Before each release, you will do a 2-week pre-release Sprint of final integration, testing, etc.

Velocity data from first 3 sprints is given below. These 3 sprints are over. The last sprint completed on 31-Dec-2010.

	Backlog Item	Estimated Size
Sprint 1	Implement the ___ functionality	20
	Fix the ___ bug	50
	Rearchitect the ___ module	20
	Investigate the ___ solution	60
	Upgrade the ___ servers	60
Sprint 2	Implement the ___ functionality	50
	Fix the ___ bug	60
	Rearchitect the ___ module	20
	Investigate the ___ solution	20
	Upgrade the __ servers	10
	Implement the ___ functionality	10
Sprint 3	Fix the ___ bug	40
	Rearchitect the ___ module	60
	Investigate the ___ solution	40
	Upgrade the __ servers	40
	Implement the ___ functionality	40
	Fix the ___ bug	20

2011 Calendar

Mon	Tue	Wed	Thu	Fri
02-Jan	03-Jan	04-Jan	05-Jan	06-Jan
09-Jan	10-Jan	11-Jan	12-Jan	13-Jan
16-Jan	17-Jan	18-Jan	19-Jan	20-Jan
23-Jan	24-Jan	25-Jan	26-Jan	27-Jan
30-Jan	31-Jan	01-Feb	02-Feb	03-Feb
06-Feb	07-Feb	08-Feb	09-Feb	10-Feb
13-Feb	14-Feb	15-Feb	16-Feb	17-Feb
20-Feb	21-Feb	22-Feb	23-Feb	24-Feb
27-Feb	28-Feb	01-Mar	02-Mar	03-Mar
06-Mar	07-Mar	08-Mar	09-Mar	10-Mar
13-Mar	14-Mar	15-Mar	16-Mar	17-Mar
20-Mar	21-Mar	22-Mar	23-Mar	24-Mar
27-Mar	28-Mar	29-Mar	30-Mar	31-Mar
03-Apr	04-Apr	05-Apr	06-Apr	07-Apr
10-Apr	11-Apr	12-Apr	13-Apr	14-Apr
17-Apr	18-Apr	19-Apr	20-Apr	21-Apr
24-Apr	25-Apr	26-Apr	27-Apr	28-Apr
01-May	02-May	03-May	04-May	05-May
08-May	09-May	10-May	11-May	12-May
15-May	16-May	17-May	18-May	19-May
22-May	23-May	24-May	25-May	26-May
29-May	30-May	31-May	01-Jun	02-Jun
05-Jun	06-Jun	07-Jun	08-Jun	09-Jun
12-Jun	13-Jun	14-Jun	15-Jun	16-Jun
19-Jun	20-Jun	21-Jun	22-Jun	23-Jun
26-Jun	27-Jun	28-Jun	29-Jun	30-Jun
03-Jul	04-Jul	05-Jul	06-Jul	07-Jul
10-Jul	11-Jul	12-Jul	13-Jul	14-Jul
17-Jul	18-Jul	19-Jul	20-Jul	21-Jul
24-Jul	25-Jul	26-Jul	27-Jul	28-Jul
31-Jul	01-Aug	02-Aug	03-Aug	04-Aug
07-Aug	08-Aug	09-Aug	10-Aug	11-Aug
14-Aug	15-Aug	16-Aug	17-Aug	18-Aug
21-Aug	22-Aug	23-Aug	24-Aug	25-Aug
28-Aug	29-Aug	30-Aug	31-Aug	01-Sep
04-Sep	05-Sep	06-Sep	07-Sep	08-Sep
11-Sep	12-Sep	13-Sep	14-Sep	15-Sep
18-Sep	19-Sep	20-Sep	21-Sep	22-Sep
25-Sep	26-Sep	27-Sep	28-Sep	29-Sep
02-Oct	03-Oct	04-Oct	05-Oct	06-Oct
09-Oct	10-Oct	11-Oct	12-Oct	13-Oct
16-Oct	17-Oct	18-Oct	19-Oct	20-Oct

Mon	Tue	Wed	Thu	Fri
23-Oct	24-Oct	25-Oct	26-Oct	27-Oct
30-Oct	31-Oct	01-Nov	02-Nov	03-Nov
06-Nov	07-Nov	08-Nov	09-Nov	10-Nov
13-Nov	14-Nov	15-Nov	16-Nov	17-Nov
20-Nov	21-Nov	22-Nov	23-Nov	24-Nov
27-Nov	28-Nov	29-Nov	30-Nov	01-Dec
04-Dec	05-Dec	06-Dec	07-Dec	08-Dec
11-Dec	12-Dec	13-Dec	14-Dec	15-Dec
18-Dec	19-Dec	20-Dec	21-Dec	22-Dec
25-Dec	26-Dec	27-Dec	28-Dec	29-Dec

Product Backlog as of today i.e. Jan 1 2011.

Id	Backlog Item	Estimated Size	Cumulative total
1.	Implement the ___ functionality	20	20
2.	Fix the ___ bug	10	30
3.	Rearchitect the ___ module	40	70
4.	Investigate the ___ solution	10	80
5.	Upgrade the ___ servers	10	90
6.	Implement the ___ functionality	50	140
7.	Fix the ___ bug	60	200
8.	Rearchitect the ___ module	40	240
9.	Investigate the ___ solution	10	250
10.	Upgrade the __ servers	40	290
11.	Implement the ___ functionality	60	350
12.	Fix the ___ bug	60	410
13.	Rearchitect the ___ module	30	440
14.	Investigate the ___ solution	40	480
15.	Upgrade the __ servers	30	510
16.	Implement the ___ functionality	20	530
17.	Fix the ___ bug	10	540
18.	Implement the ___ functionality	10	550
19.	Fix the ___ bug	60	610
20.	Rearchitect the ___ module	10	620
21.	Investigate the ___ solution	50	670
22.	Upgrade the ___ servers	60	730
23.	Implement the ___ functionality	10	740
24.	Fix the ___ bug	20	760
25.	Rearchitect the ___ module	60	820
26.	Investigate the ___ solution	30	850
27.	Upgrade the __ servers	60	910
28.	Implement the ___ functionality	50	960
29.	Fix the ___ bug	60	1020
30.	Rearchitect the ___ module	10	1030

31.	Investigate the ___ solution	10	1040
32.	Upgrade the __ servers	40	1080
33.	Implement the ___ functionality	10	1090
34.	Fix the ___ bug	20	1110
35.	Implement the ___ functionality	20	1130
36.	Fix the ___ bug	20	1150
37.	Rearchitect the ___ module	10	1160
38.	Investigate the ___ solution	50	1210
39.	Upgrade the ___ servers	20	1230
40.	Implement the ___ functionality	30	1260
41.	Fix the ___ bug	30	1290
42.	Rearchitect the ___ module	20	1310
43.	Investigate the ___ solution	30	1340
44.	Upgrade the __ servers	20	1360
45.	Implement the ___ functionality	30	1390

Exercise – Self Organization of Team

You are the ScrumMaster. What would you do under the following circumstances?

1. The team has four developers, two testers and a DBA. Developers and Testers are not working well together. Developers work in isolation. Two days before the end of sprint, they throw the code “over the wall” to testers.
2. The team is failing to deliver potentially shippable software at the end of sprint. Nothing is 100% production grade. They are close, but work needs to be done on the same user story in the next iteration.
3. The team seems to be consistently under committing during sprint planning. They finish the work that they commit, but it does not seem much. The Product owner has not yet complained about this issue.
4. The organization has all its code in Java. The team likes to work on new hot technologies. The team has chosen Ruby and Rails for development. The Product owner is not technical and cares only about the application running.
5. The organization has 20 agile projects. Each team has its own testers, who are starting to go in different directions in terms of preferred tools and approaches.
6. Jeff, a senior developer, is very dominating. During iteration planning, the team defers to him on every decision even though he is a horrible estimator. You notice the glances that other team members exchange when he suggests low estimates on some user stories.
7. You are the ScrumMaster for two teams. One team discusses all sides of various issues raised before making a decision. With the other team discussions drag on endlessly because they pursue absolute consensus in all cases.

Exercise – TeamWork Survey

Objectives:
To identify the present stage of the teamwork model that your team is presently operating in.

Directions

Below is mentioned a questionnaire that contains statements about teamwork. Next to each question, indicate how often your team displays each behavior by using the following scoring system:

- Almost never - 1
- Seldom - 2
- Occasionally - 3
- Frequently - 4
- Almost always - 5

Part 1 - Questionnaire

1. _____ We try to have set procedures or protocols to ensure that things are orderly and run smoothly (e.g. minimize interruptions, everyone gets the opportunity to have their say).
2. _____ We are quick to get on with the task on hand and do not spend too much time in the planning stage.
3. _____ Our team feels that we are all in it together and shares responsibilities for the team's success or failure.
4. _____ We have thorough procedures for agreeing on our objectives and planning the way we will perform our tasks.
5. _____ Team members are afraid or do not like to ask others for help.
6. _____ We take our team's goals and objectives literally, and assume a shared understanding.
7. _____ The team leader tries to keep order and contributes to the task at hand.
8. _____ We do not have fixed procedures, we make them up as the task or project progresses.
9. _____ We generate lots of ideals, but we do not use many because we fail to listen to them and reject them without fully understanding them.
10. _____ Team members do not fully trust the others members and closely monitor others who are working on a specific task.
11. _____ The team leader ensures that we follow the procedures, do not argue, do not interrupt, and keep to the point.
12. _____ We enjoy working together; we have a fun and productive time.
13. _____ We have accepted each other as members of the team.
14. _____ The team leader is democratic and collaborative.
15. _____ We are trying to define the goal and what tasks need to be accomplished.
16. _____ Many of the team members have their own ideas about the process and personal agendas are rampant.
17. _____ We fully accept each other's strengths and weakness.

18. _____ We assign specific roles to team members (team leader, facilitator, time keeper, note taker, etc.).
19. _____ We try to achieve harmony by avoiding conflict.
20. _____ The tasks are very different from what we imagined and seem very difficult to accomplish.
21. _____ There are many abstract discussions of the concepts and issues, which make some members impatience with these discussions.
22. _____ We are able to work through group problems.
23. _____ We argue a lot even though we agree on the real issues.
24. _____ The team is often tempted to go above the original scope of the project.
25. _____ We express criticism of others constructively
26. _____ There is a close attachment to the team.
27. _____ It seems as if little is being accomplished with the project's goals.
28. _____ The goals we have established seem unrealistic.
29. _____ Although we are not fully sure of the project's goals and issues, we are excited and proud to be on the team.
30. _____ We often share personal problems with each other.
31. _____ There is a lot of resisting of the tasks on hand and quality improvement approaches.
32. _____ We get a lot of work done.

Part 2 - Scoring

Next to each survey item number below, transfer the score that you give that item on the questionnaire. For example, if you scored item one with a 3 (Occasionally), then enter a 3 next to item one below. When you have entered all the scores for each question, total each of the four columns.

Item	**Score**	**Item**	**Score**	**Item**	**Score**	**Item**	**Score**
1.	________	2.	________	4.	________	3.	________
5.	________	7.	________	6.	________	8.	________
10.	________	9.	________	11.	________	12.	________
15.	________	16.	________	13.	________	14.	________
18.	________	20.	________	19.	________	17.	________
21.	________	23.	________	24.	________	22.	________

27. ________	28. ________	25. ________	26. ________
29. ________	31. ________	30. ________	32. ________
TOTAL _____	**TOTAL _____**	**TOTAL _____**	**TOTAL _____**
Forming Stage	**Storming Stage**	**Norming Stage**	**Performing Stage**

This questionnaire is to help you assess what stage your team normally operates. It is based on the "Tuckman" model of Forming, Storming, Norming, and Performing. The lowest score possible for a stage is 8 (Almost never) while the highest score possible for a stage is 40 (Almost always).

The highest of the four scores indicates which stage you perceive your team to normally operates in. If your highest score is 32 or more, it is a strong indicator of the stage your team is in.

The lowest of the three scores is an indicator of the stage your team is least like. If your lowest score is 16 or less, it is a strong indicator that your team does not operate this way.

If two of the scores are close to the same, you are probably going through a transition phase, except:

- If you score high in both the Forming and Storming Phases then you are in the Storming Phase
- If you score high in both the Norming and Performing Phases then you are in the Performing Stage

If there is only a small difference between three or four scores, then this indicates that you have no clear perception of the way your team operates, the team's performance is highly variable, or that you are in the storming phase (this phase can be extremely volatile with high and low points).

Exercise – Teamwork

Take a coin. Choose a partner. Your partner also has a coin. Both you and your partner choose either head or tail from you coin, without showing the coin to your partner. Both show their choice of the coin side at the same time to each other. Each person calculates his/her score by using the table:

Coin Side	Points	Comments
Head Tail	-5 +5	Both the person have different choice
Head Head	+1 +1	Both have chosen head
Tail Tail	-1 -1	Both have chosen tail

Repeat the above sequence multiple times. Keep a total of your score. The game will have a fixed time. You can change your partner at any time. You may also stop playing. The goal is to have maximum score at the fixed time.

Exercise – Requirements in Waterfall vs. Iterative.
We have a bag of coins. It needs to be sorted and total amount of money computed. Give your lowest bid in terms of time. The person with the lowest bid wins the contract and has to perform in the given time period.

Exercise – We are having a Party

Ingredients:

- At least 10 pages (8.5' X 11') per participant
- 1 marker per participant
- Stickers

Directions:

We are having a party, and we need to enlist everybody in the room to create invitation cards (3 per person). Begin by showing an example of what the finished card should look like:

1. fold page in half
2. draw a happy face on the front,
3. write a message on the inside,
4. sign the card,
5. stamp the back (sticker) and,
6. mail the card by dropping it in a box.

Once everybody is comfortable with all of the steps, start the timer and have participants build 3 cards each by completing each step to completion before moving on to the next step; this is known as batch & queue. Stop production about half way through and ask everybody what would happen if we decided to change the color of the paper. How much wasted effort would there be? How does this map to software? Let production continue and note the time when the first card is delivered to the customer and again when all cards are complete.

Run the process again. This time, have participants complete a card before moving on to the next; this is known as single piece flow. Again, stop production about half way through and ask the same questions as before. Let production continue and compare the times with the first method. Obviously, the second method is much faster at getting something to the customer, but more surprisingly, the second method is also faster over all.

Discuss why this is; if the participants say that it is because they have become more efficient, then run it again with the first method and challenge them to beat their time.

Learning Points:

- By taking a smaller set of requirements all the way to completion, you get something to the customer faster. Conversely, if all the requirements are processed at the same time, changes later in the cycle become more costly.

- Single piece flow is often faster than batch and queue. This is due to the fact that each cross-functional participant can take ownership of a module all the way to completion, reducing overall task-switching and hand-offs.

Exercise – Football Scrum

Timing: 15 mins

Ingredients:

- Football (or some other kind of ball)

Directions:

In order to enforce the rules of the daily stand-up meeting, a football can be used. Only those holding the football can speak. Once they have completed answering their questions, they can throw the football to another team member who has not yet spoken. This continues until all members have spoken. It is up to each individual to remember who has not spoken. You can implement penalties for violations of the rules (scoreboard, $1 to the happy hour or lunch fund, etc.)

Learning Points:

- Tracking who has not yet spoken and the expectation of receiving the football keeps everybody alert and involved.
- Only one member speaks at a time.

Exercise – You are in control (not)

Timing: 30-45 mins

Ingredients:

- Four unique paper airplane instructions, one set for each team.
- A big stack of standard size white papers and a smaller stack of yellow papers.

Directions:

In teams of 4 or more, have participants create as many paper airplanes as possible. When thrown from behind a table at one end of the room, airplanes must cross the room and touch the opposite wall. The facilitator, playing the role of the customer, can reject any planes that do not meet their quality standards. Track the number of planes created/approved, time to get the first plane approved, time to absorb a new team member, time to incorporate a new requirement (first yellow plane) First pass: Self organizing and cross functional teams No roles or responsibilities. No prep time. Provide each team one paper airplane instruction, the same for all teams. Go! Second pass: command and control with specialists. Create new teams. Team members may only perform one function:

- Folder: Can fold paper so that the surfaces remain in contact.
- Bender: Can bend the paper into a new angle, so long as the surfaces are not touching (this would be the job of the Folder).
- Pilot: May do final adjustments to elevators.

A team member may change their role; however they must leave for 1 minute to attend training. A Project Manager is in charge of overall quality. They must establish the steps and tasks for each team member. Give the team 5 minutes to prepare their process. Go! Run exercise for 5-10 minutes for each pass. After two-four minutes, swap one team member from each team, After three - six minutes, put in a special order for yellow planes. Offer bonus of ten points per yellow plane.

Learning Points:

- Self organizing and cross functional teams are better able to adapt to changes in than those driven by command and control with specialists.
- In addition, they are faster at getting to market and more productive.

Exercise – The Story of Sprints

Timing: 10 mins

Ingredients:

- People and space
- 1 stop watch
- Optionally, something to record the audio with.

Directions:

Have the team sit/stand in a circle. You want to get a story of the last sprint that is told by the entire team. You start by saying 'Once upon a time , we had a X (insert sprint length here!) week sprint...'. Then, the next person to your left adds to your sentence and this carries on until the last person has spoken or if the story is developing in an interesting direction, until all the points appear to have been made and there is nothing of value coming through. You might want to strictly enforce the time limit for a large team.

After the retrospective , you could run the game again to tell the story of the next sprint, and this should galvanise the improvements that will take place and nicely summarise the lessons learnt and help the team visualise how the next sprint could be better. This game helps to create an ongoing shared goal and represents an oral history of the software process.

Learning Points:

- Discover a consensus view of the success/failures from the last sprint.
- Empower everyone to add value to a collective goal through participation.
- Exercise the 'responding to change' learning point from the word-at-a-time letter game.

Exercise – Ball Point Game

Playing the Game
In order to play the Ball Point game, you’ll need a large open space with enough room for everyone to stand. You’ll also need about 20 brightly colored tennis balls and you may want a whiteboard to do the debriefing. We play the game in the following way:

1. Provide an overview of the game and the rules.
   - Everyone is part of one big team.
   - Each ball must have air-time.
   - Each ball must be touched at least once by every team member.
   - Balls cannot be passed to your direct neighbor to your immediate left or right.
   - Each ball must return to the same person who introduced it into the system.
   - There are a total of five iterations.
2. Allow the team two minutes of preparation time to determine how they will organize themselves.
3. Get an estimate from the team of how many balls they can pass through the system.
4. Run a two-minute iteration.
5. Allow the team one minute to discuss how to improve the process.
6. Repeat for five iterations. Make the fifth iteration a challenge.

The objective of the Ball Point game is to get as many balls through the team as possible within two minutes. Each ball must be touched at least once by every team member and must end with the same person with whom it began. After two minutes the team is allowed an additional minute to discuss the process and how it could be improved. The game is played a total of five times.

Initially, the Scrum Trainers will have difficulty getting a single ball through the system, but, after adapting our process, the interactions became more predictable and the participants started to make visible progress. It usually takes four cycles for participants to reach a point where they were both productive and having fun.

The fifth (and final) iteration is a challenge Sprint. The Trainer challenges the participants to get as many balls through the system as possible. Usually the participants will reorganize themselves.

Learning: All systems have a natural velocity and if we want to increase this natural velocity, we need to change the boundaries of the system.

Exercise – Spec Writer
Duration: 30 minutes

In this team-based exercise, each team is divided into "Developers" and "Spec-writers." The "Developers" are separated from the "Spec-writers" and only allowed to communicate using written specifications. "Spec-writers" are then presented with a diagram that they need to communicate to the "Developers," who, in turn, must interpret the written specifications and reproduce the diagram. The exercise is run twice with two different diagrams and a retrospective is held at the end of each run.

1. Find two "Developers" in each team. (I usually do that by asking who is a PM or non-developer since I want them to feel how hard it can be to understand a specification.)
2. The rules of the exercise are:
    - a) The originals cannot leave the room.
    - b) Specifications must be written. Diagrams, symbols and numbers are not permitted.
    - c) As many specifications as desired can be delivered, as often as desired.
    - d) The only allowed communication is for "Spec-writers" to hand over the written specifications to developers.
    - e) "Spec-writers" can look at what the "Developers" are doing, but not communicate verbally or with body language.
3. Move the "Developers" out of the room and place them so that the "Developer" teams cannot see what other teams are doing.
4. Distribute copies of the first original to the "Spec-writers." It's important that the "Developers" don't see the original.
5. Run the exercise. Often "Spec-writers" have a hard time not communicating with the "Developers" with body language, so pay attention to this.
6. After 12 minutes stop the teams and collect the results. Show them to class.
7. Let the teams do a retrospective under the given rules.
8. Run the exercise again with the second original.

Diagrams: http://kanemar.files.wordpress.com/2008/03/oppgaveomkravformidling1.pdf

Learning Points: It show the difficulties that the "Developers" have understanding the written specifications and the importance of verbal communication in communicating a vision. It also demonstrates some of the difficulties that teams have when there is a separation between the "Spec-writers" and the "Developers."

The separation of "Spec-writers" and "Developers" greatly reduces the team's performance.

Exercise – Play word association

The goal of the game is to provoke thought. This is not a test, not a test of our skill.

Given the list of the words on the left, find the most appropriate match on the right. Since this a game, even though both “ATDD” and “TDD” are listed on the left, we won’t find “Testing” anywhere on the right.

ATDD	Bad
Backlog	Coordinate
Daily Standup / Daily Scrum	Design
Product Owner	Facilitate Collaboration
ScrumMaster	Iteration
Sprint	Specify
TDD	To Do List
WIP	Tradeoffs

The End.
Please send your feedback on this document to vijay_nathani@yahoo.com. Thank you.