scrum

“Man, the living creature, the creating individual, is always more important than any established style or system”

“If you spend too much time thinking about a thing, you'll never get it done. “

“absorb what is useful, discard what is useless…”

BRUCE LEE

TABLE OF CONTENTS

INTRODUCTION 5
- **Background** 5
- **Manifesto for Agile Software Development** 6

WHAT IS SCRUM? 7
- **Summary** 7
- **Roles and Responsibilities of a Scrum Team** 8
  - Product Owner 8
  - ScrumMaster 8
  - Team Member 8
- **The Size of a Scrum Team** 9

THE PREGAME PHASE 10
- **Determine Feasability** 11
- **Set The Product Vision** 12
- **Create an Initial Product Backlog (PB)** 13
  - What is the product backlog? 14
  - What information do we capture in the product backlog? 15
- **Prioritise the Product Backlog** 16
- **Estimate The Product Backlog** 17
  - Selecting an Estimation Technique 17
    - Estimate Using Ideal Time 17
    - Estimate Using Story Points 19
  - Make Estimation Fun Use Planning Poker 20
  - Adjust Estimates 21
    - Complexity Factor 21
    - Drag Factor 21
    - Working Environment 21
    - Multiple-Teams 21
    - Are there multiple teams involved? 21
    - Adjustment Factor Chart 22
  - Choose Sprint Length 23
- **Calculate Initial Velocity** 23
  - What is Velocity? 23
  - How to Calculate Initial Velocity? 23
  - Calculating Duration 24
  - Estimate an Initial Cost and Duration 24
- **Initial Release Planning** 25
- **Create High Level Architecture and Design** 25
- **Identify Risks** 25

THE GAME PHASE 26
- **The Sprint Planning Meeting** 26
  - Calculate Hours Available to the Sprint Prior to Meeting 26
  - Set the Sprint Goal 27
  - Creating a Sprint Backlog (Task Expansion) 27
- **Executing the Sprint** 28
  - Allocating Tasks (We Dont) 28
  - The Daily Stand-up Meeting 29
  - Sticking to the Sprint Backlog 30
  - Using a White-board for the Sprint Backlog 31
  - Daily Estimation of Time Remaining 31
  - The Sprint Burn-down Chart 32
  - What do we do with Product Backlog Items not Completed? 33
  - The Sprint Retrospective 34
  - Re-planning After or During a Sprint 34
  - The Product Burn-Down Chart 36
  - Parking Lot Report 38
  - Defect Report 38

POSTGAME PHASE.....39
APPENDIX.....40
Planning Poker Cards.....40
Cards 1 and 2.....40
Cards 3 and 5.....41
Cards 8 and 13.....42
Cards 21 and 34.....43
Frequently Asked Questions.....44
Special Thanks.....44

INTRODUCTION

Background

The experience after decades of following traditional heavyweight prescriptive practices has made it evident that :

- The clients or users are not sure what they want.
- They have difficulty stating all they want and know.
- Many details of what they want will only be revealed during development.
- The details are overwhelmingly complex for people.
- As they see the product develop, they change their minds.
- External forces (such as a competitor's product or service) lead to changes or enhancements in requests.

Bullet Point Source: Agile and Iterative Development: A Manager's Guide by Craig Larman

In the mid 1990's individuals were frustrated with these shortcomings and a number of alternative methods emerged. These methods have become standard practice in many companies worldwide. These new practices fall under the term "agile methods."

On February 11-13, 2001 these visionaries met and the Agile Alliance was formed.

"The **Agile Alliance** is a non-profit organization that supports individuals and organizations who use agile approaches to develop software. Driven by the simple priorities articulated in the "Manifesto for Agile Software Development" agile development approaches deliver value to organizations and end users faster and with higher quality." Source: www.agilealliance.org

The Agile family consist of a number of methods / frameworks that apply different techniques but adhere to the same principles. A popular method within this family is "scrum." Scrum was created by Ken Schwaber, Jeff Sutherland and Mike Beedle.

Scrum is a simple process and is industry proven, it is not a silver bullet and requires hard work and commitment.

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

Source: www.agilemanifesto.org

WHAT IS SCRUM?

Summary

- Scrum is an iterative, incremental process for developing any product or managing any work. It produces a potentially shippable set of functionality at the end of every sprint (iteration). It's attributes are:
- Scrum is an agile process to manage and control development work.
- Scrum is a wrapper for existing engineering practices.
- Scrum is a team-based approach to iteratively, incrementally develop systems and products when requirements are rapidly changing
- Scrum is a process that controls the chaos of conflicting interests and needs.
- Scrum is a way to improve communications and maximize co-operation.
- Scrum is a way to detect and cause the removal of anything that gets in the way of developing and delivering products.
- Scrum is a way to maximize productivity.
- Scrum is scalable from single projects to entire organizations. Scrum has controlled and organized development and implementation for multiple interrelated products and projects with over a thousand developers and implementers.
- Scrum is a way for everyone to feel good about their job, their contributions, and that they have done the very best they possibly could.

Source: http://www.controlchaos.com/about

Scrum consists of the following three phases

Roles and Responsibilities of a Scrum Team

Product Owner	The product owner (PO) represents the customer(s), stakeholder(s) or funder(s) of the product. The PO is a part the team delivering the product. The product owner is responsible for: • setting the product vision • managing the return on investment (ROI) • present initial and ongoing product requirements to the team • prioritise each requirement relative to business value • manage new requirements and their prioritisation • release planning • act as mediator when there are multiple customers • ensure subject matter experts (SME's or business domain experts (BDE's) are available to the team
ScrumMaster (Project Manager)	The project manager (PM) role differs from traditional command and control methods. The PM in scrum works with and for the team. The PM is responsible for: • allowing the team to self-organise to get the work done • ensure that communication paths are open and efficient • ensure and help the team to follow the scrum process • remove any impediments (issues) that the team encounter • protect the team from outside interferences to ensure that their productivity is not affected • facilitate the daily stand-up meeting
Team Member	A team member is an individual that commits to getting the work done. Team members are cross-functional. What do we mean by that? In traditional methods we have distinct roles e.g. tester or architect. In a scrum team we have people with architecture skills, testing skills, however many individuals for example an architect could have sufficient secondary skill to help out in testing. These cross-functional skills help the team to get the work done. The team member is responsible for: • setting the sprint goal • committing to the work and doing it to the highest quality • work to the product vision and sprint goal • collaborating with team members and help the team to self-organise • estimating of requirements and ensuring that the work effort remaining is up to date • attending the daily stand-up meeting • raise impediments

The Size of a Scrum Team

A scrum team should not exceed 9 people (+- 2, 7 is the recommendation.) Small teams are more productive than large teams, one reason is the number of communication paths within a team.

If the team is larger than this recommended size then it should be split into 2 teams and synchronisation should occur.

The formula below allows us to calculate the number of communication paths in a team where everyone is **equal**. I emphasize the word equal as this is purely an indicative measurement as in the real world it is not equal e.g. Management Hierarchies (Program Director, Product Director etc.)

N is the number of people in a team.

Communication Paths **=** [*N* * (*N*-1)] / 2

We get:

Number of People in Team	Number of Communication Paths
1	0
5	10
10	45
15	105
20	190
25	300
30	435

THE PREGAME PHASE

The pregame phase is akin to the traditional "initiation and planning" phases. We apply a lightweight approach to ensure that we get to producing working product features quickly. During the pregame phase we perform the following:

- Determine Feasibility
- Set the Product Vision
- Create an initial product backlog
- Prioritise the Product Backlog
- Estimate the Product Backlog
- Choose Sprint Length
- Calculate Initial Velocity
- Estimate Resources and Cost
- Create High Level Architecture and Design
- Identify Risks

The pregame phase is short it is not like a traditional waterfall approach where for example 8 months is spent writing a Business Requirement Specification, Design Specification, Software Requirements Specification, Detailed Test Strategy etc. This phase can be a matter of days or weeks. The aim is to get approval and buy-in and begin delivering product value to the customer quickly.

Determine Feasibility

In traditional methods this can take a long time, but lets take a different approach. What if we got the right people into a room and workshop the product feasibility with the aim of obtaining indicative savings or potential earnings. Most people based on an idea will already know if it is feasible or not.

The importance here is that a decision needs to be made, and made quickly. Please note that I know it is more complex but I hope that this will motivate you to see things in a simpler and lightweight way.

Lets walk through a hypothetical example for a bank and show how this can work.

"*The mortgage group based in headquarters consists of 50 mortgage administration staff who are responsible for processing mortgage applications. When a mortgage application is received the staff process it and then file it in the customer file stored in the filing room. They use an excel spreadsheet to track the status of the application as in most cases require further financial checks etc so they are deemed pending until approved or not approved.*

The mortgage group manager (Lets call her Lisa) has recommended that this is a major problem and needs to be fixed. Her staff are spending up to 2 hours a day retrieving and locating customer files, their mortgage product is popular and they are processing around 1500 a month (75 p.d) . Staff morale is low as they are only not coping with the demand (she needs an extra 4 staff at the minimum) and she is concerned that this archaic process is starting to affect customer attitude and service levels. They are known for their service and are worried that a competitor may start chipping away their customer base if service levels drop.

Lisa has recommended that this needs to move to an electronic based process. The divisional manager, team leader, herself and a few staff book out half a day to workshop the concept."

They white-board the problem and together work out some indicative metrics.

They take the average daily cost per staff member, work out an hourly rate and determine that this 2 hour waste equates to roughly $75 per day per staff member.

Waste Cost = (Staff X Waste Cost Per Day) X Number of Average Days Worked Per Year
Waste Cost = (50 X $75) X 222
Waste Cost = $832,000

They discuss this further and work out the following:

Staff processing about 1.5 applications per day, eliminating this waste the group would be able to process an extra 18 applications per day, an extra 360 a month.
Eliminates new hiring of 4 staff (around $266,000 per annum)
Eliminating waste is the equivalent of around 12 FTE's (about $800,000 labour cost)
Customer service levels improve

The divisional manager is excited and everyone agrees that this is very feasible. He is willing for the group to spend some time work out some costs and high level requirements and he will present this to the national director for funding approval if it looks like a winner.

Set The Product Vision

The product vision is our elevator pitch and allows everyone to understand the big picture.

This vision statement guides product development activities and provides direction to the team on what we are trying to achieve. The vision statement allows us to open ourselves to the fact that we can achieve the possible.

The following areas may help to direct you towards creating a product vision:

- People (e.g. customers, users)
- The Organization
- Quality
- Innovation and Creativity (e.g. market differentiation)
- Environment (e.g. competitors, partners)
- Perception

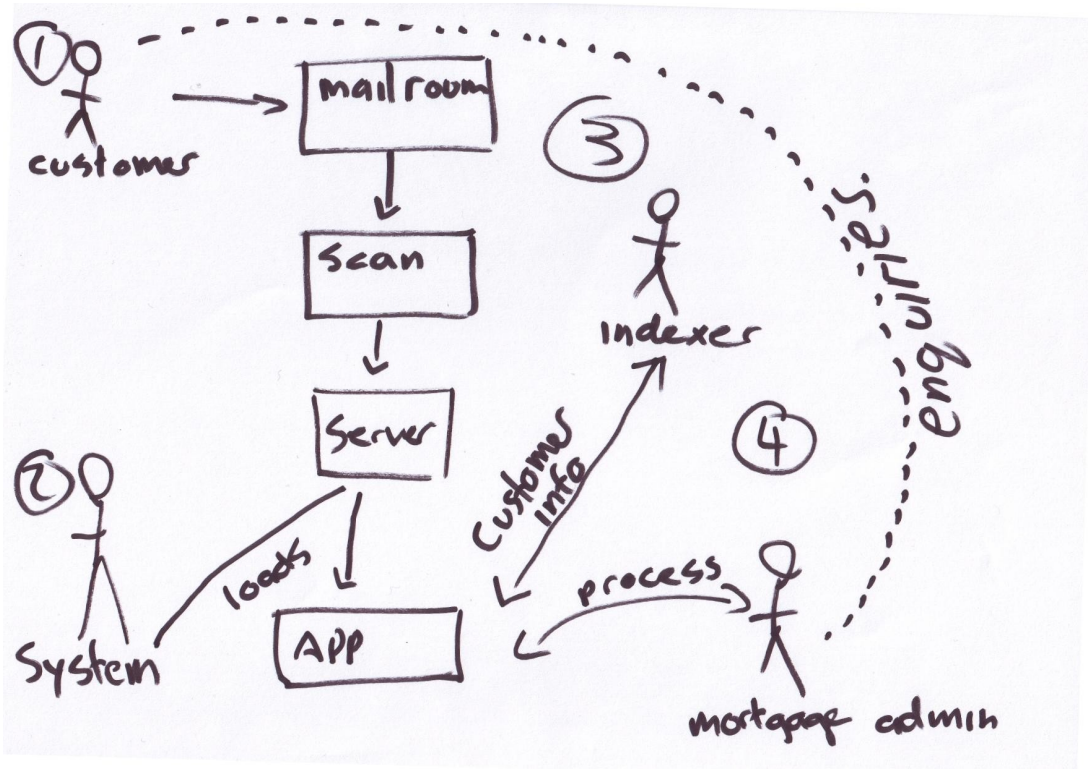
Using our previous example Lisa and her team sit down for 2 hours an workout a product vision. They come up with the following:

"To implement centralised electronic document repository for mortgage applications providing the bank with the ability to handle increases in mortgage applications without hiring new staff through productivity gains. The solution will allow us to reduce our customer inquiry time-frames maintaining our competitive advantage as a customer focused provider of financial services, at the same time staff attrition will be reduced providing significant savings in replacing and training existing staff."

Create an Initial Product Backlog (PB)

Lisa and representatives from her team book another workshop, invite some IT participants and spend half a day identifying requirements at a very high level. They are now ready to populate an initial product backlog.

Another hypothetical example of one of their white-board efforts is show below:



What is the product backlog?

The product backlog is owned and managed by the Product Owner.

It is a dynamic prioritized source for all the work that needs to be done to deliver a product. This artifact is product focused not system focused.

A product backlog consists of:

- Features
- Enhancements
- Defects
- Non IT Deliverables

The list of items that reside within the product backlog are known as Product Backlog Items (PBI's.) These items are high level descriptions of work known as user stories. Examples are shown below:

- As the mail room I must scan mortgage applications from the customer so that they can be processed
- As the system I must load scanned images so they are available to the indexing user
- As the indexer I want to be able to view a list of all applications requiring indexing
- As the indexing user I want to be able to record customer information for a mortgage application
- As the marketing department I want to produce a product brochure to hand-out at our internal conference

Some people prefer to use a shorter form for example:

- Scan mortgage application forms
- Load images and make available to indexer
- View list of applications requiring indexing
- Indexer record customer information against application
- Produce Marketing Product Brochure

For more information on user stories please refer to the following sources:

http://www.mountaingoatsoftware.com/article/view/27
http://www.extremeprogramming.org/rules/userstories.html

What information do we capture in the product backlog?

The details below provide a guideline to what information you may want to capture in a product backlog.

Item	Description
Reference	Unique identifier for quick reference during collaboration
Description	A high level description of the item
Type	Feature, New Feature, Enhancement, Defect
Group	This can also be known as a feature set but is simply a grouping of functionality for example: Customer Management Contact Management
Relative Value	A factor that identifies the business value (Scale of 1 to 10, 10 been the highest business value)
Initial Estimate	This can be in story point or ideal days
Adjustment Factor	A summation of factors that may increase the initial estimate
Actual Estimate	Initial Estimate X (1 + Adjustment Factor)
Release	The release number that the PBI will be included in
Sprint (Iteration)	The sprint that the PBI has been allocated to
Notes	Additional comments, references or points of interest, be careful not to get into too much detail, some people prefer this to be separate
Status	Done or Not Done
Committed	Has this item been committed by the product owner i.e it has been decided that it will be incorporated into the product

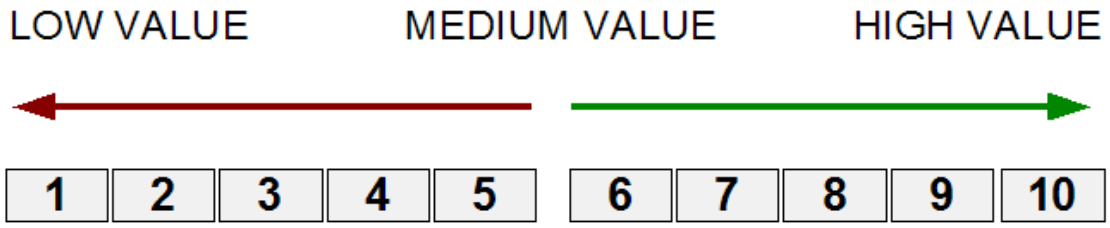
An example is shown below, in all examples I am assuming that all items have been committed.

Ref	Description	Type	Group	Relative Value	Initial Estimate	Adjust Factor	Estimate	Release	Sprint	Status
01	As the mail room I must scan mortgage applications from the customer so that they can be processed	Feature	Scanning and Imaging	10	3	0	3	1	1	Not Done
02	As the system I must load scanned images so they are available to the indexing user	Feature	Scanning and Imaging	10	5	0.5	7.5	1	1	Not Done

We are not concerned that the PBI's are not detailed. Customer's do not know all their requirements up-front and customers change their minds when they see working features. We keep the PBI's at a high level, this informs us that we need to collaborate with the customer when we start working on it. Allowing the customer to decide as late as possible gives us an advantage as they will have the most up to date information at hand and we are more likely to deliver a product that meets their business needs.

Prioritise the Product Backlog

Every item on the product backlog needs to be prioritised. This prioritisation tells us which items we should be delivering first. To prioritise a product backlog item the product owner determines the relative business value of the item using the scale below.



Here are a few guidelines to assist in determining the business value of the product backlog item.

- What is the financial benefit of me introducing this feature? Will it increase revenue or is the development effort and cost more than it's really worth?
- Will this feature increase productivity and usability or will it cause frustration and resentment towards the product and organisation?
- If I introduce this feature will it result in large training costs for our users or customers?
- If I introduce this feature will it result in a change in business processes? If yes are we taking into account the cost of that change?
- What is the penalty if I do not implement this feature? e.g. Legislation and compliance
- If I introduce this feature will it result in an increase in risk or will it eliminate a number of evident risks we are currently exposed to?
- Will this feature differentiate us in the market?

The backlog excerpt below shows us an example of prioritised items:

Ref	Description	Type	Relative Value
01	Add a new customer	Feature	10
02	Edit an existing customer	Feature	10
03	Remove an existing customer	Feature	3
04	Add a customer contact	Feature	9
05	Edit an existing customer contact	Feature	9
06	Remove an existing customer contact	Feature	3
07	View company location in street directory map	Feature	1

WHAT DOES THIS TELL US?

The customer wants us to deliver the ability to add and edit customer information first.

Then they want us to deliver the ability to add and edit the contacts for a customer.

Then they want us to provide the capability to delete customers and contacts.

Lastly they want us to integrate with street directory maps to view the customer location

Estimate The Product Backlog

What is an estimate?

Estimate

1 an approximate calculation.
2 a written statement indicating the likely price that will be charged for specified work.
3 a judgment or appraisal

Source: Oxford Dictionary

To estimate the product backlog items we follow this simple process:

SELECT ESTIMATION TECHNIQUE ➔ INITIAL ESTIMATION OF ITEMS ➔ ADJUST ESTIMATES

Selecting an Estimation Technique

Estimate Using Ideal Time

We might estimate that something will take x days of effort to complete, unfortunately this is not how long it will take (duration.)

For example:

John estimates that to complete item ABC that it will take him 2 days of effort.

On day one an urgent support call for a production system needs resolution, John spends half a day resolving it and spends half a day working on item ABC.

On day 2 on the way to work, John receives a call that his mother has fallen down the stairs, he rushes off to ensure she is ok, he spends a day away from work at the hospital.

John returns on day 3 and does half a day of work on ABC, this half a day was not very productive as he was constantly thinking about his mom. In the afternoon he shoots off to pickup his mom as she is been discharged from hospital after an overnight precautionary stay.

John returns on day 4 and spends the whole day on ABC he is productive.

On day 5 John spends half a day and completes item ABC

John estimated that it will take 2 days of effort to complete item ABC
It actually took John 2.5 days of effort to complete item ABC
The duration that it took to complete item ABC was 4.5 days

It is not possible to take into account all the interruptions that team members will experience and accurately estimate how long something will take to do.

Another example:

John might take 2 days of effort to complete something, Mary is not as experienced as John and may take 4 days. What happens if John is injured and off for the duration of the project and Mary is brought in to cover for John, in theory all of John's work is going to take twice as much effort.

This is why we estimate in ideal days or hours. This is how long we think something would take if their were no interruptions or obstacles to complete a product backlog item. This should be a team based estimate where possible.

Example:

John, Mary and Paul sit down. They are required to calculate the estimate for the product backlog items. For each item the individuals estimate in ideal time how long an item would take to complete. This can either be averaged as shown below or use the same technique in the next section on Story Points.

Item to Estimate	John	Mary	Paul	Estimate in Ideal Days
As a site visitor I want to be able to subscribe to the company newsletter	6	3	4	4.3
As the system I must send out a confirmation email when a site visitor successfully registers for the newsletter	1	5	3	3
As a manager I want to be able to view a report on all new newsletter subscribers for a given day	4	5	3	4
TOTAL				**11.3 days**

Estimate Using Story Points

Story points is an estimating technique that has become favorable in agile teams. Its focus is to estimate a product backlog item based on relative magnitude (size) and complexity / difficulty. There are many different types of scales that are used to allocate points. As humans we are much better at predicting size and complexity than duration.

A story point is an abstract concept.

The benefit with using story points is that it is not time based so does not require high maintenance like ideal time, if something is big and complex in most cases it will stay big and complex. It is more efficient to estimate in story points than ideal time. For example an experienced developer and a novice developer can quickly agree whether something is big and complex.

We can simply use a number series from the Fibonacci sequence. We have selected to use the values below as story points. 34 is the most complex and largest value.

1	2	3	5	8	13	21	34

In short the team members doing the work sit down together and each person estimates the size and complexity of the feature using the scale above, they discuss it, compare it to other stories we have already estimated (this is known as triangulation) and then re-estimate it. The team then reaches consensus on what value to allocate.

Example of Story Point Estimating:

The team are presented with the following feature to estimate. Each individual thinks about it and comes up with the following estimates.

Item to Estimate	John	Mary	Paul
As a site visitor I want to be able to subscribe to the company newsletter	13	5	8

John is asked why he has given a factor of 13 as there is a large variance from the others.

John states that "we will need to write a registration page, validation rules and database calls so its pretty big and there may be some complexity in it. I also think that it is roughly 3 times bigger than story X estimated earlier."

Mary says to John, "I have given it a rating of 5 because I was thinking of using the library we used last year for customers who want to register to receive our company announcements. We can re-use a lot of it, and deliver this feature early. I think its roughly about the same size as story X and Y."

Paul states "I have done this heaps of times for other companies, its pretty simple so thought I would give it an 8, but Mary I agree if you are confident that we can re-use that library it would save us a lot of time, yes I agree that it is about the same as story Y."

The team re-estimate with the following results:

Item to Estimate	John	Mary	Paul
As a site visitor I want to be able to subscribe to the company newsletter	8	5	5

The team agree to allocate the higher of the story points and select 8 as the estimate.

Make Estimation Fun Use Planning Poker

A great effective fun way in estimating is to use a concept called planning poker. There is no gambling, you can leave your wallets behind. This meeting should be time-boxed so as to allow roughly 2-3 minutes per story. Dont allow it to become a situation of 15 minutes of discussion on requirements for a story.

It is very simple:

- Each person is given a pack of cards containing the Fibonacci series (refer to Estimate Using Story Points.)
- The person who is facilitating the estimation reads out the user story (product backlog item) and any additional notes.
- The team may ask the Product Owner (or representative) a few questions. The point is make sure it is high level and does not go into detail.
- When questions are completed or time allocated time has run out each person selects their estimate (a card) based on the size and complexity of the story and places it face down on the table. The participants must refrain from indicating what they have chosen until the cards are turned face-up.
- When everyone has selected a card, they all at once turn them over.
- If there is great variance between the selected cards, the teams discusses the low and high cards and find out what those team members were thinking to get to their estimate. We are not trying to undermine their estimate but to try and find out if they were thinking of things we had not thought of or might have missed. This is a good opportunity to take notes for later.
- Everyone picks up their card and returns it to the pack. They re-estimate the story by thinking about it for a short time, selecting their card and placing it face down on the table again.
- This is repeated until the cards are close to full alignment on the estimate
- Some people put a restriction on the number of rounds to prevent getting bogged down
- The team reaches agreement on the estimate

A pack of planning poker cards can be found in appendix A

Adjust Estimates

Our initial estimates were "ideal world" estimates. Adjusting the estimate allows us to be aware of factors that are going to affect the teams productivity. Adjustment factors should be continuously reviewed. For example after three months of working together the the team should be working well together and their technology and business understanding should be sufficient.

The information below is not an accurate measure but simply allows us to realise or take into account that a lot of unknowns are going to affect the project.

An adjustment factor can be applied in the following ways:

- Calculated as a single factor and applied to a number of items
- Calculated as a single factor to be applied to a group of stories
- Calculated individually for each product backlog item

To adjust our estimates we use the following formula:

Estimate **=** *Initial Estimate* **X** (1 + (*complexity* + *drag* + *working environment* + *multiple teams*))

Complexity Factor
How complex are the requirements and technology that we are using?
Drag Factor
The drag factor takes into account three areas that will affect the teams productivity: Has the team worked together before? If yes then for how long? What level of skill does the team possess regarding the technology been used? Does the team know the business domain?
Working Environment
Is the team located together as a group or are they in separate floor areas, offices or regions? A team co-located in a single room for example is going to be more productive.
Multiple-Teams
Are there multiple teams involved? Multiple teams require synchronisation, this will affect productivity.

Adjustment Factor Chart

1. COMPLEXITY FACTOR

Simple	0
Slightly Complicated	0.1
Complicated	0.2
Very Complicated	0.6

Please note that the Complexity Factor has been adapted from the original source

2. DETERMINE WORKING ENVIRONMENT FACTOR

If the team is not co-located in one dedicated area (e.g. room) then add 0.1

3. DETERMINE MULTIPLE TEAMS FACTOR

If there are multiple teams then add 0.1

4. DETERMINE THE DRAG FACTOR

Drag Factor	Team Together	Technical Knowledge	Business Knowledge
0.8	< 3 months	Low	Low
0.75	< 3 months	Low	Medium
0.70	< 3 months	Low	High
0.75	< 3 months	Medium	Low
0.50	< 3 months	Medium	Medium
0.50	< 3 months	Medium	High
0.75	< 3 months	High	Low
0.50	< 3 months	High	Medium
0.35	< 3 months	High	High
0.60	< 1 Year	Low	Low
0.55	< 1 Year	Low	Medium
0.50	< 1 Year	Low	High
0.55	< 1 Year	Medium	Low
0.30	< 1 Year	Medium	Medium
0.25	< 1 Year	Medium	High
0.50	< 1 Year	High	Low
0.25	< 1 Year	High	Medium
0.20	< 1 Year	High	High
0.50	> 1 Year	Low	Low
0.45	> 1 Year	Low	Medium
0.40	> 1 Year	Low	High
0.45	> 1 Year	Medium	Low
0.35	> 1 Year	Medium	Medium
0.20	> 1 Year	Medium	High
0.40	> 1 Year	High	Low
0.20	> 1 Year	High	Medium
0.00	> 1 Year	High	High

Extracted from Estimation in Agile Projects, Dr. Christoph Steindl (IBM)

Choose Sprint Length

A sprint does not exceed 4 weeks. It is recommended that a sprint be set at 4 weeks (20 business days.) Others prefer 2 week sprints.

Whatever length you decide on the important thing to remember is to keep the length constant as this provides stability to the team. Do not end up in the situation below:

Sprint	Length (Duration)
1	5 Days
2	15 Days
3	10 Days
4	5 Days
5	20 Days

Calculate Initial Velocity

What is Velocity?

Velocity is the measure of the productivity of the team i.e. The the rate at which the team is completing work from the product backlog. Velocity is the most important metric as it tells us how much work can be done in a sprint.

Velocity can be used in ideal time or story point format.

How to Calculate Initial Velocity?

When we start a project, we do not know what our velocity is, we have to somehow work out an initial estimated velocity that we think we can achieve in the first sprint. When we are bit into our first sprint we will begin to realize our velocity as features begin to be completed.

The following options can be used to calculate an initial velocity:

- If you are lucky to have previous projects that have used scrum then use their actual velocity for similar work and team skill or composition. If it is the same team then even better.
- Predict the velocity, take some stories, expand them into tasks to work out what you think you can complete
- Execute a short sprint to complete a few stories to work out velocity

Calculating Duration

So how do we calculate duration?

Using story points:

Duration = Size Remaining / Velocity

Using ideal time:

Duration = Ideal Time Remaining / Velocity

Estimate an Initial Cost and Duration

Please note I am keeping this very simple and not trying to go into details of costing models etc.

We simply use this formula:

Cost of Project = Money Spent + (Sprints Remaining X Sprint Burn Rate)

I will use story points in this example but the same applies to ideal time if you choose to use it. We are going to use sprints of 20 business days.

After we have adjusted our estimates our product backlog has a total estimation of 90 story We determine that our initial velocity for sprint 1 (20 Days) will be 30. As we have not started the project our work remaining is the sum of all estimated story points i.e. 90.

So using our formula:
Duration = Size Remaining / Velocity
Duration = 90 / 30
Duration = 3

So we estimate that it will take us 3 sprints to complete the current backlog, i.e. end of March.

Jan	Feb	Mar
30	30	30

At a high level we know that our team will roughly burn around $20,000 per month, so the initial estimated cost is:

Cost of Project = Money Spent + (Sprints Remaining X Sprint Cost)
= $0 + (3 X $20,000)
= $60,000

We have now got an indicative cost to do the current work on the product backlog. We can now apply a 20% buffer to this and say that we think to deliver the product based on the initial backlog we are looking in the region of $48,000 to $72,000. In most companies the buffer margin will differ and in some they may not even use a range.

This is enough for us to approach our funder for the approval or release of initial funding.

Initial Release Planning

{to be completed}

Create High Level Architecture and Design

{to be completed}

Identify Risks

{to be completed}

THE GAME PHASE

The pregame phase has been completed and finance (or other) have release all or some of the funding to commence with delivering value. This is where the fun begins.

The Sprint Planning Meeting

The sprint planning meeting is a short meeting of around a half to a full day of duration and why not, remember we are only looking at 20 days of work. This meeting is held just before to executing a sprint. The product owner, scrum-master and team attend this meeting. The product owner attends so that he can answer any questions that the team has.

Calculate Hours Available to the Sprint Prior to Meeting

The scrum-master and team need to work out the resource availability for the sprint prior to holding the sprint planning meeting.

This is simply the number of staff hours of work available for the 20 days. A persons full day of productivity should be set at 6 hours. No-one is productive for a full day (normally 8 hours) as we take into account personal phone calls, answering emails, meetings and other interruptions.

The resource matrix below provides an illustrative example of how we can determine the hours available to a sprint.

RESOURCE PLANNING MATRIX																				
	Hours Available for each Day in Sprint																			
Resource	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Dopey	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Bashful	6	6	6	6	**0**	**0**	**0**	6	6	6	6	6	6	6	6	6	6	6	6	6
Doc	6	6	6	6	**0**	6	6	6	**0**	6	6	6	6	6	6	6	6	6	6	6
Sneezy	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Grumpy	6	6	6	6	6	**0**	**0**	**0**	**0**	**0**	6	6	6	6	6	6	6	6	6	6
Happy	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Sleepy	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
TOTAL	**42**	**42**	**42**	**42**	**30**	**30**	**30**	**36**	**30**	**36**	**42**	**42**	**42**	**42**	**42**	**42**	**42**	**42**	**42**	**42**
TOTAL NUMBER OF HOURS AVAILABLE TO SPRINT IS **780**																				

Bashful has 3 days off
Doc as 2
Grumpy is off on leave for the second week in the sprint

You may have a team member who is also responsible for third level support so you might cut their daily hours down to 4 to cater for this. We now know that we have around 780 hours of tasks that we can complete in the sprint. We are now prepared for our sprint planning meeting.

Set the Sprint Goal

The product owner based on the value prioritisation of items in the product backlog will present to the team what he or she would like delivered in the sprint. Everyone discusses and establishes an initial sprint goal. This goal is a very simple description. For example:

"Allow people to subscribe to our corporate newsletter and un-subscribe if they do not wish to receive it anymore"

This allows the team to focus on what needs to be achieved at the end of the sprint. During the course of the planning meeting this goal might be adapted if the team cannot commit to all items required to achieve the goal. The team also might add to the goal if they feel they can commit to more items.

It is important to not set bad goals e.g. "Complete Story A,B,C,D" or "Allow subscribers to do all the things they can on the product backlog", these dont serve any purpose, the team must be able to relate to these.

Creating a Sprint Backlog (Task Expansion)

The team is now ready to create the sprint backlog.

The team selects the high priority items from the product backlog that they think they can complete in the sprint and achieve the sprint goal. These items are then expanded into tasks in an artifact called the sprint backlog. The sprint backlog can be in scrum specific tool, Excel, a white-board or cards stuck on the wall.

The team works through the selected items one at a time. For each item the team discusses and determines all the tasks that are required for the item to be "done." This is known as task expansion. Remember that all sound software engineering practices are carried out in the sprint e.g. Analysis, design, coding, testing, bug fixing, user interface design etc.

It is important to understand what "done" means. Done means fully tested working software that can be potentially released if asked to.

We do not want to end up in a situation where we have partially done items, this can occur if we do not complete all product backlog items in the sprint. An item that is 80% complete is not done.

For each task the team collaborates and works out how long they think it will take (in hours) to complete. We should try and keep our task estimate within a 4 to 16 hour range. If we begin to exceed this then it is a sign that the task needs to be split in smaller tasks. Any tasks under 4 should be grouped with other tasks.

For example:

Story	Task	Est	Remaining	Who?	Status
As a site visitor I want to be able to subscribe to the company newsletter	Write subscribe acceptance test with customer	4			Not Done
	Design subscribe web page with customer	4			Not Done
	Establish fields, design and create subscription database table	8			Not Done
	Write subscribe unit test	4			Not Done
	Develop subscription CGI script	5			Not Done
	Write subscription validation rules	5			Not Done
	Design Theme for templates	4			Not Done
	Customer UAT	4			Not Done
	TOTAL	**38**			

The team have estimated that the item above will take 38 hours to complete. We do the above exercise for each item and have a running sum of the number of hours we have chewed up so far. This allows us to determine if we have selected too much or too little work.

If the team cannot deliver what the product owner requires then the goal may be adjusted, the important thing here is that all parties collaborate to reach an agreement. Once this is finalised the team has committed to do their best to deliver the items selected and to achieve the sprint goal.

For example:

Lets say we select 10 items from the product backlog. We have expanded each item into its corresponding tasks and team estimation has been completed. The total so far is 420 hours of work. Our sprint has 600 hours available. We have 180 hours of work left. We would then discuss with the product owner that we can commit to some more work and repeat the process. This may result in us changing our sprint goal.

If we selected too much work we would discuss with the product owner to return some work to the product backlog and adjust the sprint goal accordingly.

Executing the Sprint

On completion of the sprint planning meeting we are ready to execute our tasks to produce working software. During the sprint the team collaborates amongst themselves, they collaborate with the customer and use the customers feedback to ensure that they are delivering what the customer wants.

Allocating Tasks (We Dont)

In traditional project management the project manager assigns the tasks to the resources available. This is normally done upfront. In an agile environment the team is left alone to self-organise, this does not imply anarchy against management but simply the fact that they are the people doing the work and they are best left to decide on how to tackle something, when to tackle something and who should do things so they can achieve their goal at the end of the sprint.

The team collaborates and decides who should do which tasks. The team does not allocate all tasks at the beginning but as tasks are completed team members communicate and sign up to new tasks.

The Daily Stand-up Meeting

Every day the team and scrum-master meet for a daily meeting (daily scrum). It is recommended that this meeting should be held at the same place and time each day, this way we dont have team members running around trying to find out where todays meeting is.

This meeting is short and must not exceed 15 minutes. It is important to understand that the meeting is for the team and scrum-master. Observers are welcome to attend but they must not interrupt the meeting by making comments etc. The team are responsible for achieving the sprint goal, not the observer.

This meeting has the following benefits:

- visibility amongst the group, everyone knows whats going on
- allows the scrum-master to note impediments so that he / she can resolve them for the team to ensure that there productivity is not compromised in anyway

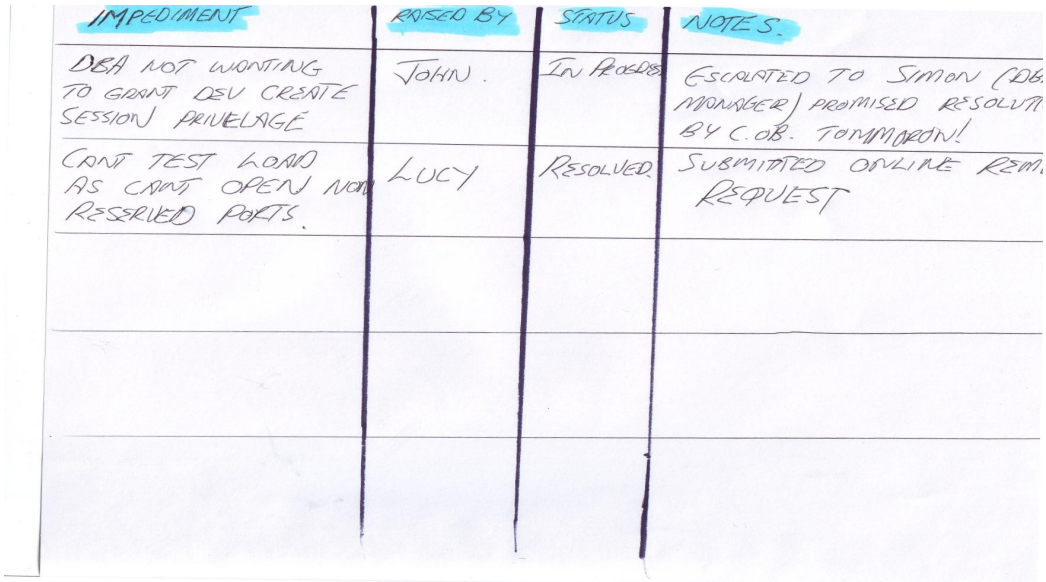
During the meeting each team member has a turn to answer the following three questions:

- What have I done since our last meeting?
- What am I doing till our next meeting?
- What is impeding me from doing my work?

The scrum-master must ensure that the team does not digress and for example start discussions on problem solving, anything that is outside the above three questions must be discussed outside of the daily meeting.

It is recommended that the scrum-master announces that the meeting is officially over so that everyone knows it is now appropriate for non meeting related items to be discussed etc.

It is the Scrum-masters responsibility to act on these impediments and resolve them for the team. Impediments prevent the team from been productive and if not resolved will cause the team to potentially not meet their sprint goal. If the scrum-master cannot resolve the impediment then they must escalate it for resolving at a higher level, in the end do whatever is possible to resolve these in the teams best interest. I recommend that a register of impediments be kept. This register can be in any form be it excel or a white-board. An example is shown below:



Sticking to the Sprint Backlog

When a sprint is running the scrum-master should avoid introducing or removing product backlog items. This is to ensure that the team focuses and completes the work they have committed to. This should only be done if too little or too much work has been committed

For example:

We are in week one of our 4 week sprint. A request has come to the product owner that the business want a new feature added and worked on now. The product owner comes to the team and tells them they must work on this new feature.

This is disruptive to the team and introduces instability. The team has spent time planning the sprint, the team is mentally focused on reaching their goal and more importantly the team is already working on delivering software.

The product owner should communicate this to the customer to try and get them to understand how this will disrupt the team. The best outcome is to prioritise it as high value on the product backlog to be worked on in the next sprint.

Common sense needs to be applied to this guideline.

If for example for a critical bug was found that that had major consequences for the organisation, lets say "credit card payments were been duplicated." then it would be understandable to immediately include it in the current sprint and return something else back to the product backlog.

If for some reason the situation is a total shambles then it is best to dissolve the sprint and start again with a sprint planning meeting. Ensure that a sprint retrospective is held if termination does occur you want to be able to find out the real causes for this event happening. Remember this is a last resort.

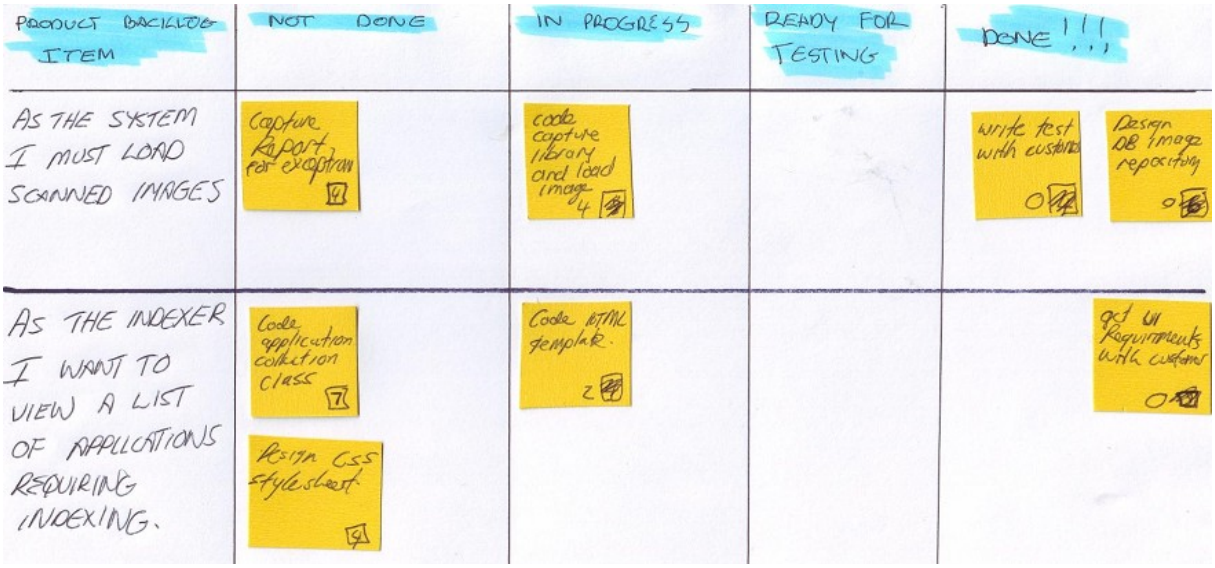
If this occurs (it shouldn't) then a multitude of factors may be causing it, here are a few for consideration:

- poor planning
- prioritisation is not correct
- lack of understanding and support of agile practices
- business environment has had massive change
- product owner is a cowboy or scrum-master is too soft
- customer is not spending enough effort on the product
- corporate politics

Using a White-board for the Sprint Backlog

It is highly recommended that a white-board or wall be used for the sprint backlog. This provides a visual medium for the team to see what is going on and who is working on what. It also assists with the daily stand-up meeting and progress reporting.

The layout below shows how the medium can be structured, the yellow cards are the tasks:



Not Done – Shows us the work that has not yet been started

In Progress – Shows us what the team is currently working on

Ready for Testing – Needs testing by testers or customer

Done – Task has been fully tested and is complete

When a developer selects a card to work on they move it into the "IN PROGRESS" column, when they have completed it they move it into the "READY FOR TESTING" column. When the testers have completed testing and it has passed they move it to the "DONE" column.

At the end of each day the team member will write down the estimated time remaining on the task card not in the "Not Done" and "Done" columns, this feeds the sprint burn-down chart.

Daily Estimation of Time Remaining

At the end of each day each team member updates how much time is remaining for their task they are currently working on. If they do finish something then they will set the number of hours remaining to 0. This is a very quick and simple mechanism yet it provides so much value.

The Sprint Burn-down Chart

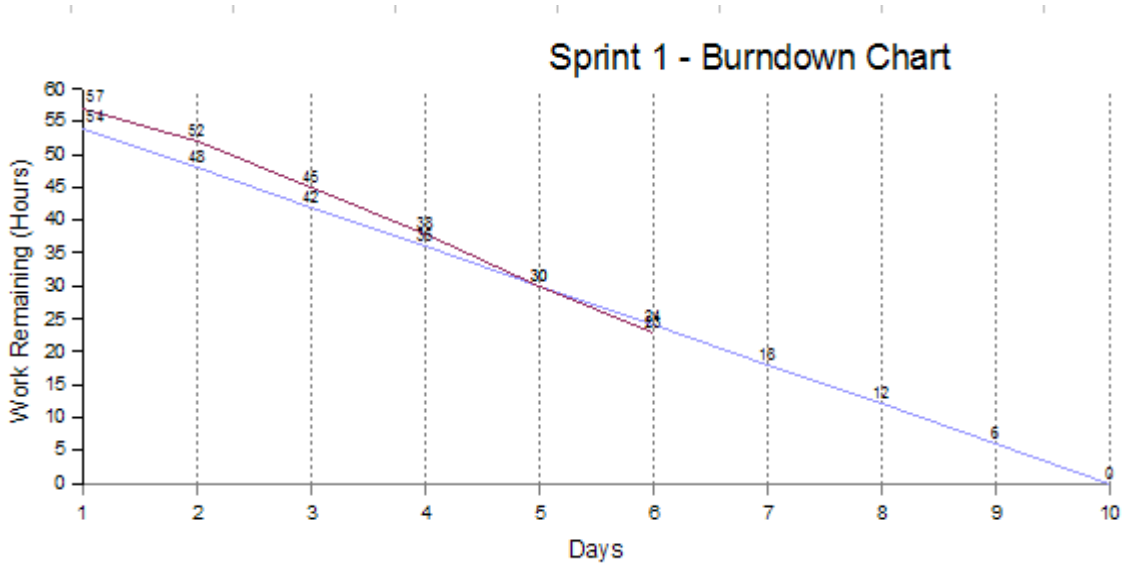
The sprint burn-down chart provides us with a daily snapshot on the teams velocity i.e. Their productivity. It allows us to track the teams progress through the sprint and if the team is going to meet their goal that they have committed to. We simply plot the work remaining (Y Axis) vs time (X Axis.) and add an ideal velocity line.

The chart aids us in decision-making such as:

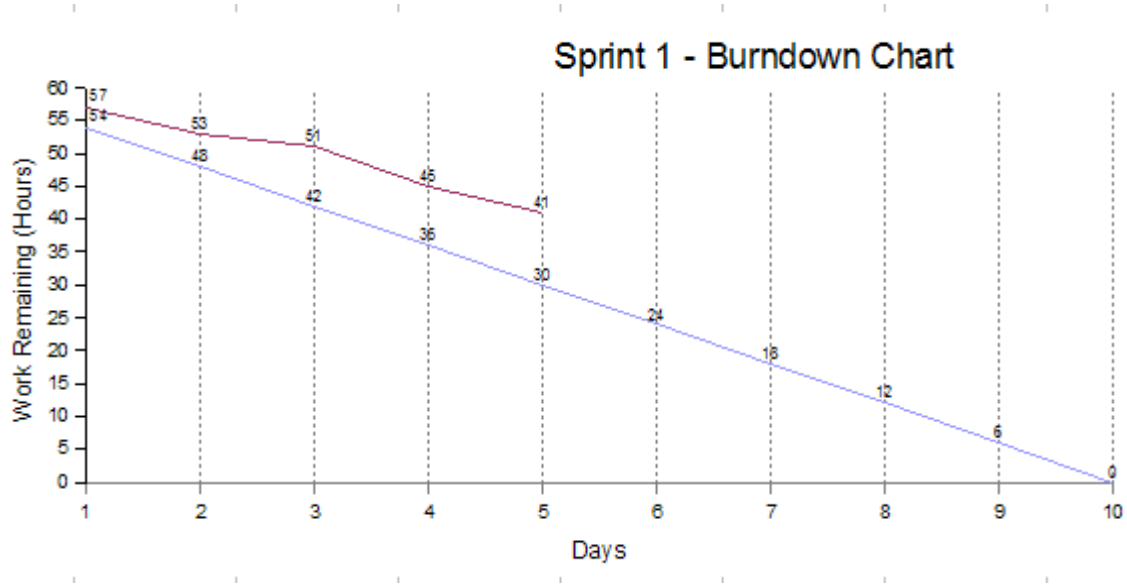
- Is the team heading towards their goal
- Do we need to pull more work into the sprint i.e. We are churning through the work, have not committed enough work
- Do we need to pull work from the sprint i.e. We have committed to to much work

Lets look at some examples:

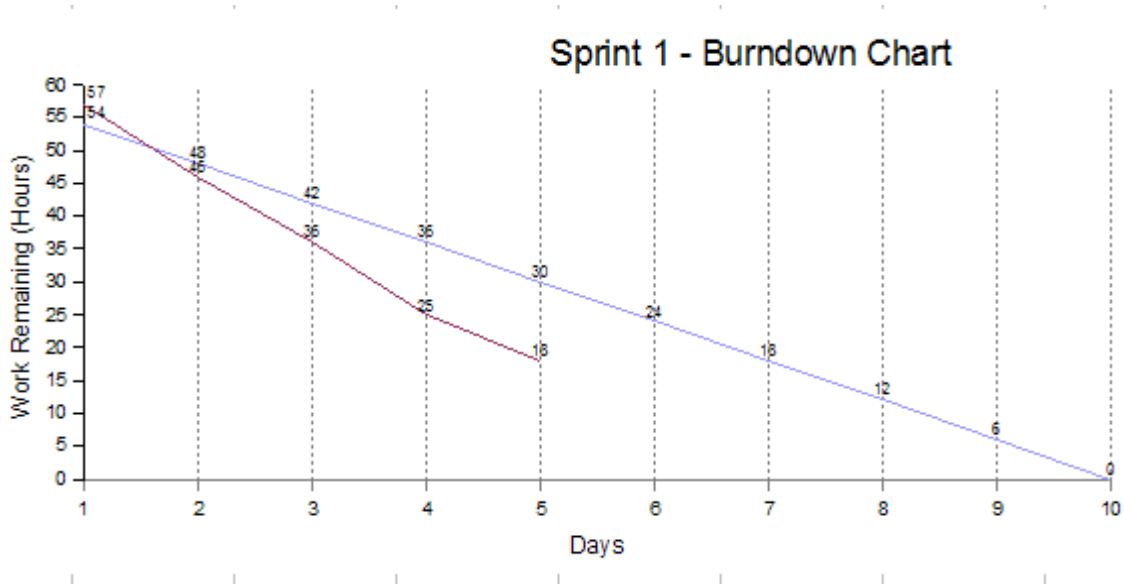
Things are looking good, the team is progressing nicely.



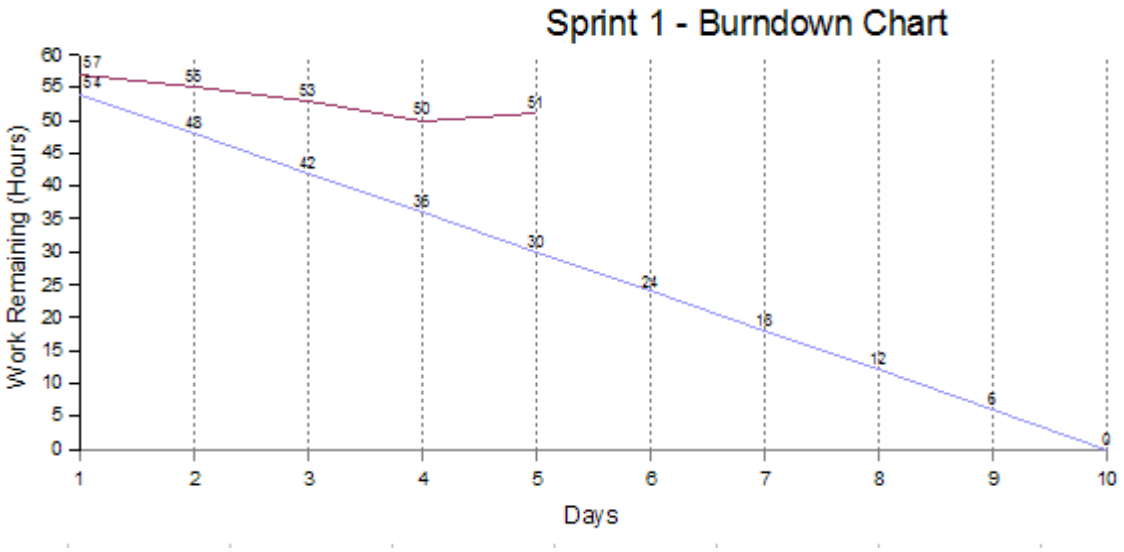
Things are not good, we need to remove some work.



Things are too good, we need to pull in some more work



We are in a lot of trouble here



What do we do with Product Backlog Items not Completed?

Items that are not completed are not done. These items are returned to the product backlog. You may have completed 80%, but I repeat, it is not done. The amount of time to complete it in the next sprint will be quicker as it is 80% done, think of it as credits in the next sprint.

The Sprint Retrospective

At the end of each sprint a retrospective is held. It is attended by the product owner, scrum-master and all team members. This meeting allows the team to "inspect" the sprint that has just concluded and "adapt" so that they can continuously improve. Here are some examples on what can be covered:

- What did we do well during this sprint?
- What did we do badly and what can we do to fix it?
- How is the team's morale is everyone happy?
- Hows our communication?

Re-planning After or During a Sprint

At the start of our project our product backlog has a total size of 100 story points. We determine that our initial velocity for sprint 1 (30 Days) will be 20.

Duration = 100/20, so we estimate that it will take us 5 sprints to complete the current backlog, i.e. end of May.

At a high level we know that our team burns $20,000 per month, so the initial estimated cost is:

Initial Cost of Project = Money Spent + (Sprints Remaining X Sprint Cost)
= $0 + (5 X $20,000)
= $100,000

Jan	Feb	Mar	Apr	May
20	20	20	20	20

We have a rough estimate of how much the current items on the backlog will cost to deliver and have determined that we anticipate finishing these items around the end of May.

After iteration one our current velocity is actually 30 (i.e. completed 30 points), so size remaining is 70 (Total – Work Completed to Date.)

Duration = 70/30, So at our current rate we anticipate that it will take us another 2.3 iterations, i.e. Mid April and the projected cost is $66,000 ($20,000 + (2.3 X $20,000).)

Jan	Feb	Mar	Apr
30	30	30	10

At the end of iteration 2 our velocity was 15, the work involved was complex and we had problems with accessing some of the business resources to obtain certain information, we took this up with scrum-master during the iteration as it came out during our daily meeting, he/she indicated the cost of this problem to the product owner and it was resolved within 2 days.

Duration = 55/15, So at our current rate we anticipate that it will take us 3.6 iterations, i.e. 3rd week of June and the projected cost is $112,000 ($40,000 + (3.6 X $20,000).)

Jan	Feb	Mar	Apr	May	Jun
30	15	15	15	15	10

At the end of iteration 3 our velocity was 30.

Work remaining = 25 (55 – 30)

Duration = 25/30, So at our current rate we anticipate that it will take us 0.8 i.e. around 1 more iteration.

The projected cost is $76,000 ($60,000 + (0.8 X $20,000).)

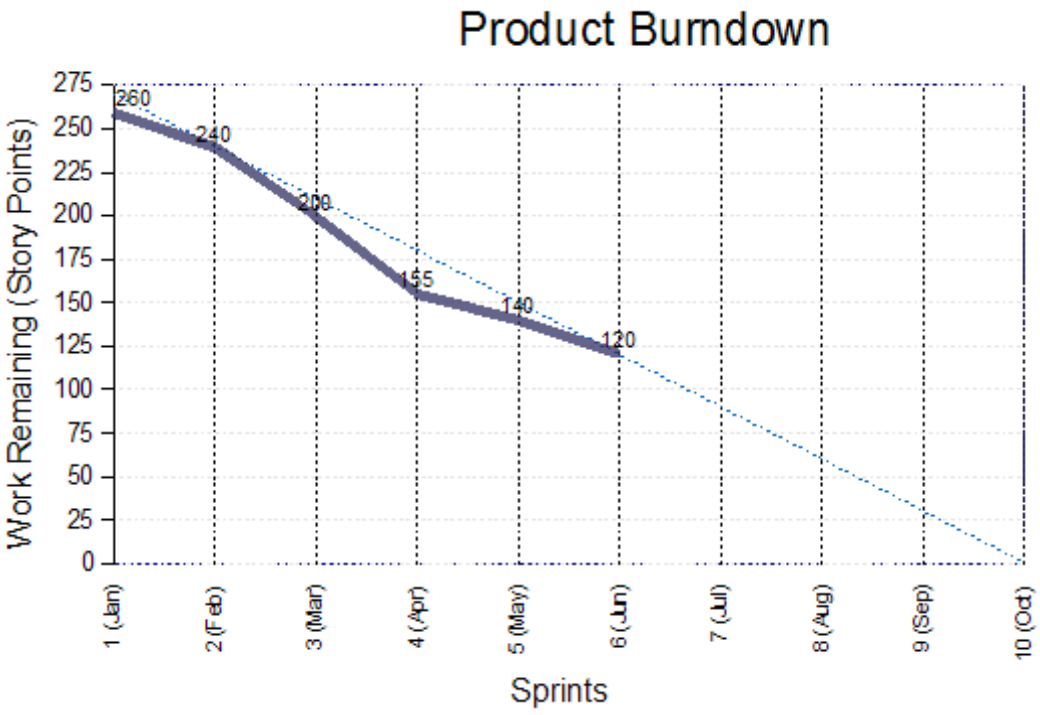
Jan	Feb	Mar	Apr
30	15	30	30

At the end of iteration 4 our velocity was 30. The cost is $80,000

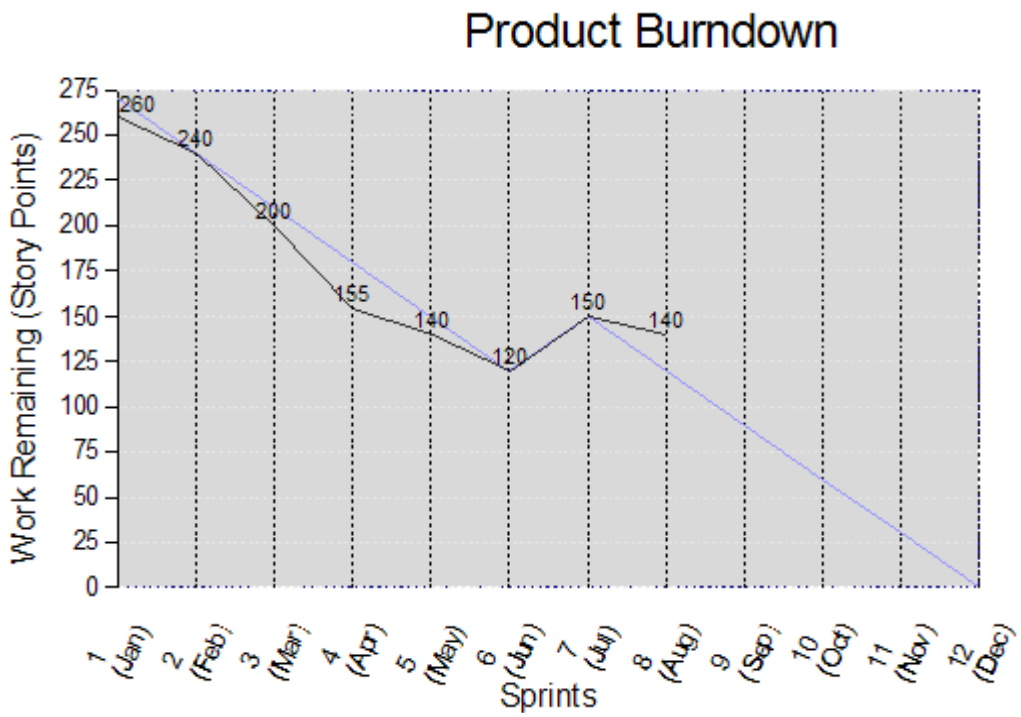
Jan	Feb	Mar	Apr
30	15	30	25

The Product Burn-Down Chart

The product burn-down chart provides provides us with a view of how the team are progressing in delivering the product. It works in the same way as the sprint burn-down chart. There are many variants of the chart, the example below shows a simple version.



Everything is answerable on the product burndown chart, for example:



Stakeholder:

"Whats going on you guys are not doing to well, I dont understand why you are lagging behind, this is costing me an arm and a leg !"

Scrum-master:

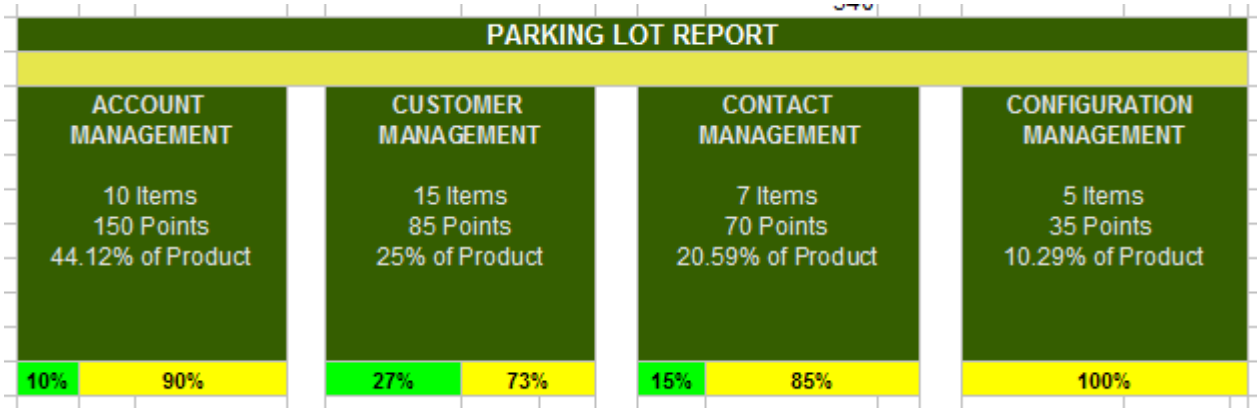
"Well its not bad, the product owner actually added 20 more features which you approved thats why it has spiked"

Stakeholder:

"Oh, sorry ..."

Parking Lot Report

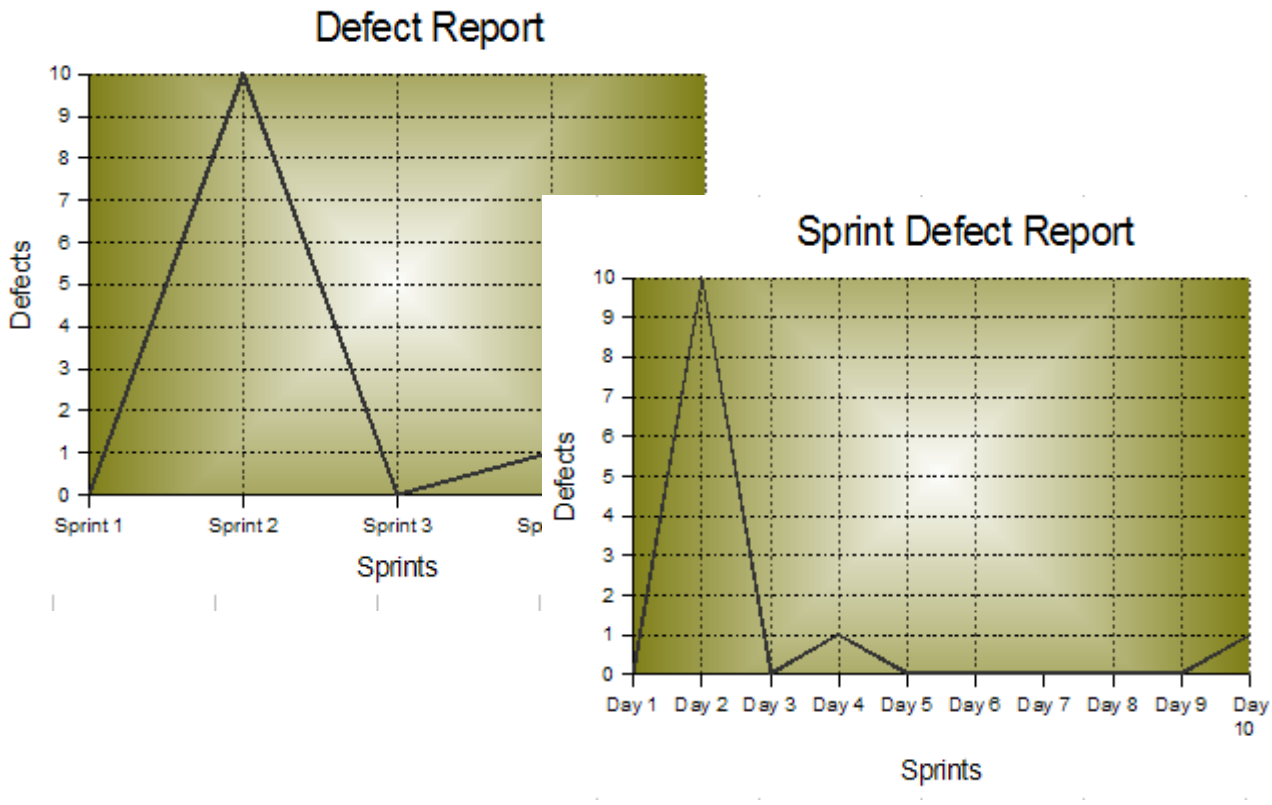
The parking lot report is a great way to show overall progress in a project via a dashboard. We use the Group (or Feature Set) from the product backlog. There are a number of ways to create parking lot reports, an example is shown below:



Lets look at Account Management:

- Account Management comprises of 10 items on the product backlog
- The total estimate for account management is 150 story points
- Account Management makes up 44.12% of the total product effort

Defect Report



POSTGAME PHASE

APPENDIX

Planning Poker Cards

Simply cut along the outline of card, fold it and cover it with contact film or wide clear tape.

Cards 1 and 2

Cards 3 and 5

Cards 8 and 13

Cards 21 and 34

Frequently Asked Questions

Special Thanks

John Cornell, Manager, Agile/Distributed Process Engineering