SWEN90016

Software Processes & Project Management

Cost Estimation

2021 – Semester 1

Tutorial 7

Today's aim

Become familiar with

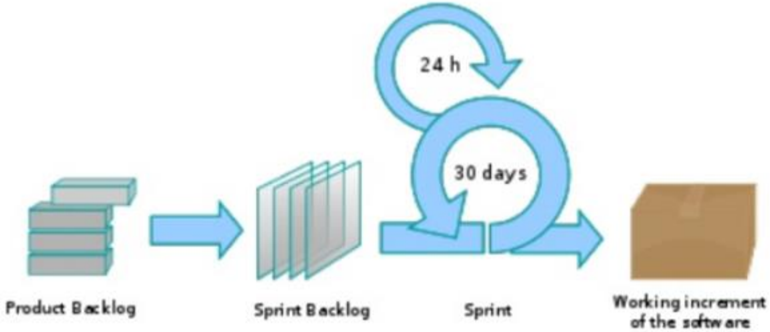
Agile

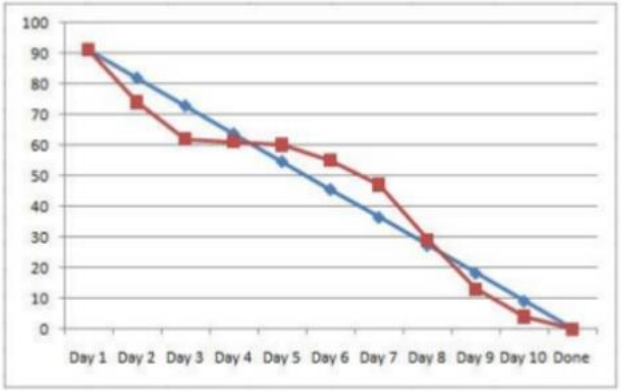
User Stories and Story Points and Velocity

Formal

Function Point Analysis and COCOMO II

Scrum Overview





Roles

Product Owner
Scrum Master
Development Team

Ceremonies

Daily Stand Up
Sprint Planning
Sprint Review
Sprint Retrospective

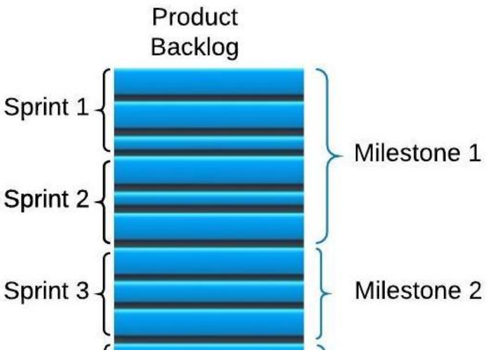
Artifacts

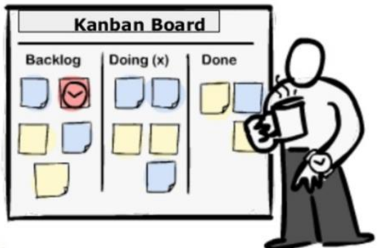
User Story
Product Backlog
Sprint Backlog
Burndown Chart
Burnup Chart

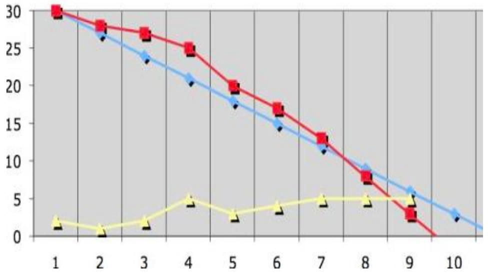
From Lecture 2, slide 51

Scrum Artifacts Overview

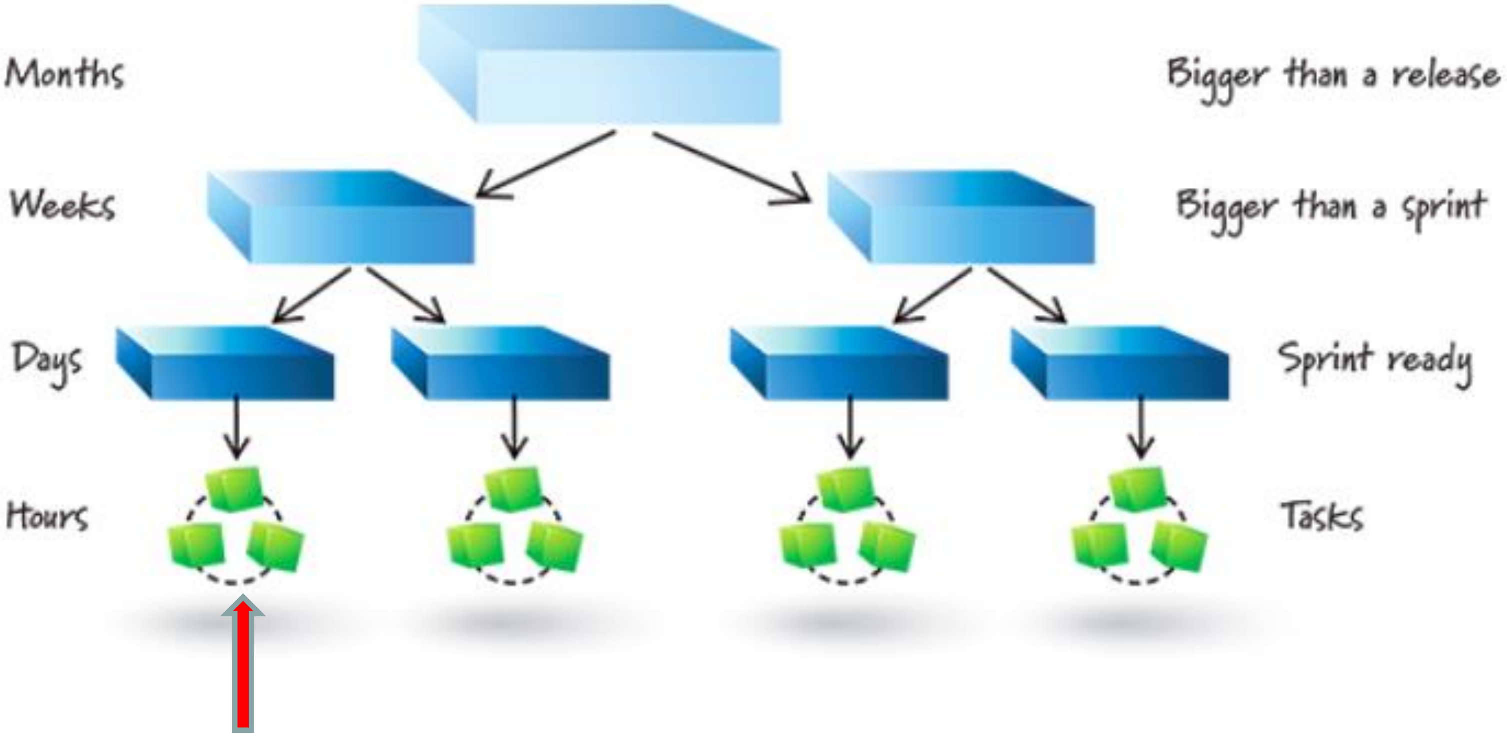
- User Stories
  - *As a <user>, I want < goal> so that < reason>.*
- Product Backlog
  - Features listed in client priority order
  - Release milestones annotated to list
- Sprint Backlog
  - Features selected for this iteration
  - Visual Kanban board
- Burn Down Chart
  - Measure the features **100% done**







User Story: revision



Product Owner has *a conversation* with the Developer to understand requirement

User Story Size

Contentious! Advice from the internet may vary …

(Sprint) User Story

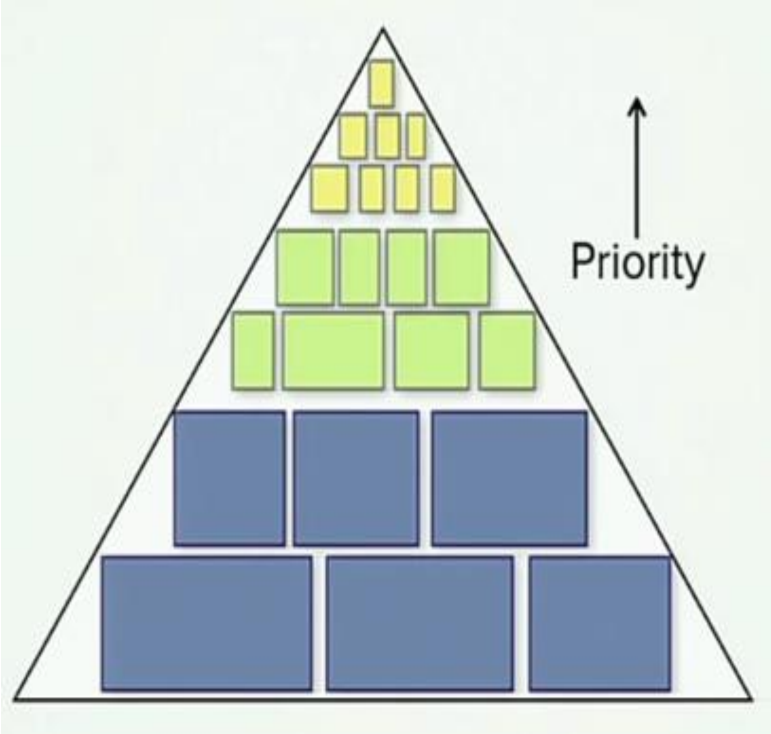
- A developer’s perspective
- A conversation placeholder

Feature User Story

- Product capabilities
- Product Owner perspective

Epic User Story

- New business services
- A product



User Story Effort Estimation

Story points: a relative measure of the size of a user story
(the requirements of the system are documented using user stories)

From Lecture 6, slide 71

raw values are unimportant

relative values matter

2 point story is twice as long as a 1 point story

limit range of Sprint Backlog estimates to 1-10

Small
1 pt
No sweat
Medium
3 pts
Nothing we can't handle
Large
5 pts
This is going to take some effort

User Story Effort Estimation

A practical Example of Size vs Duration

- I am tasked with moving a large pile of dirt from the front of my home to the back yard.
- I could look at the pile of dirt, assess my tools [a shovel and a wheelbarrow], and directly estimate the job at two hours.
- In arriving at this estimate I bypassed any estimate of size and went directly to an estimate of duration.

A practical Example of Size vs Duration

- Suppose instead that I look at the pile and estimate its size.
- Based on its dimensions I estimate the pile to contain about 100 cubic meters of dirt. This is my estimate of the size of this project.
- We want to know how long it will take to move the dirt: **Duration**
- We need to convert the estimate of size [100 cubic meters] into an estimate of duration.
- A label on my wheelbarrow says it has a capacity of two cubic meters.
- Dividing 100 cubic meters by 2 cubic meter, I decide that moving the dirt will take 50 trips with the wheelbarrow.
- I estimate that each trip will take three minutes to load the wheelbarrow, two minutes to walk to the back yard and dump the dirt, and one minute to walk back with the empty wheelbarrow. Total trip time will be six minutes.

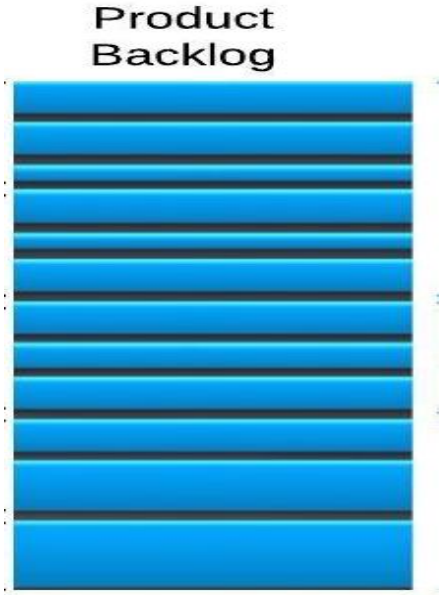
- Since I anticipate making 50 trips taking 6 minutes each, my estimate of duration is 300 minutes or 5 hours.

Scrum Release Planning

Fixed Date and Time constraints

Project:
phase
Initiation

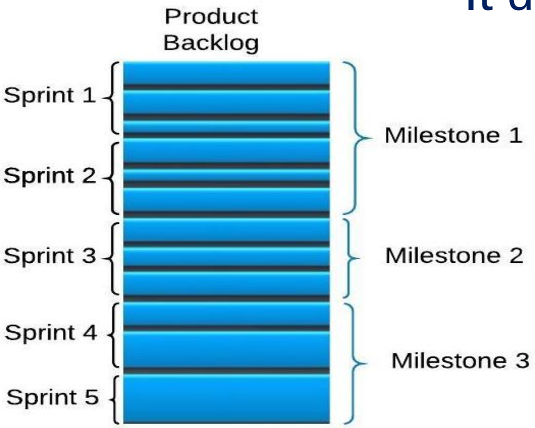
- Business Roadmap identifies candidate project
- Product vision established with external stakeholders
    - Create Product Backlog



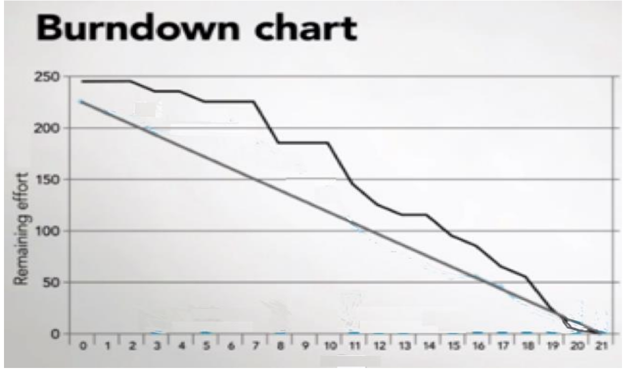
Scrum Release Planning

Project phase: Initial Sprint Planning

- The groomed Product Backlog is estimated in Story Points
  - Cheap & quick estimation
  - Low quality indicators of {easy, medium hard}
  - Let estimates have larger values, like 21 or 100 are valid
- Find the dev team's Story-Point **Velocity** measure
  - It determines the **release** schedule



Y-axis: Story Points



X-axis: Sprints

Sprint Planning

Fixed Date and Time constraints

Project phase: every Sprint Planning

- Create Sprint Backlog
  - Select high value User Stories from Product Backlog
  - Use velocity to fit appropriate number of Story Points
- Decompose selected User Stories on Sprint Backlog
  - Do ***Just-In-Time*** detailed estimation
  - Check number of Story Points will still fit
    - Detailed high quality estimation
    - Let estimates have smaller values, like 1 or 10 are valid

Humans have good judgement across one order of magnitude, but beyond that, humans are unreliable

Sprint Planning

Fixed Scope constraints

Project phase: every Sprint Planning

The User Stories can be decomposed into tasks,

- Optionally estimate tasks in hours

Less accurate www.scruminc.com/story-points-why-are-they-better-than/

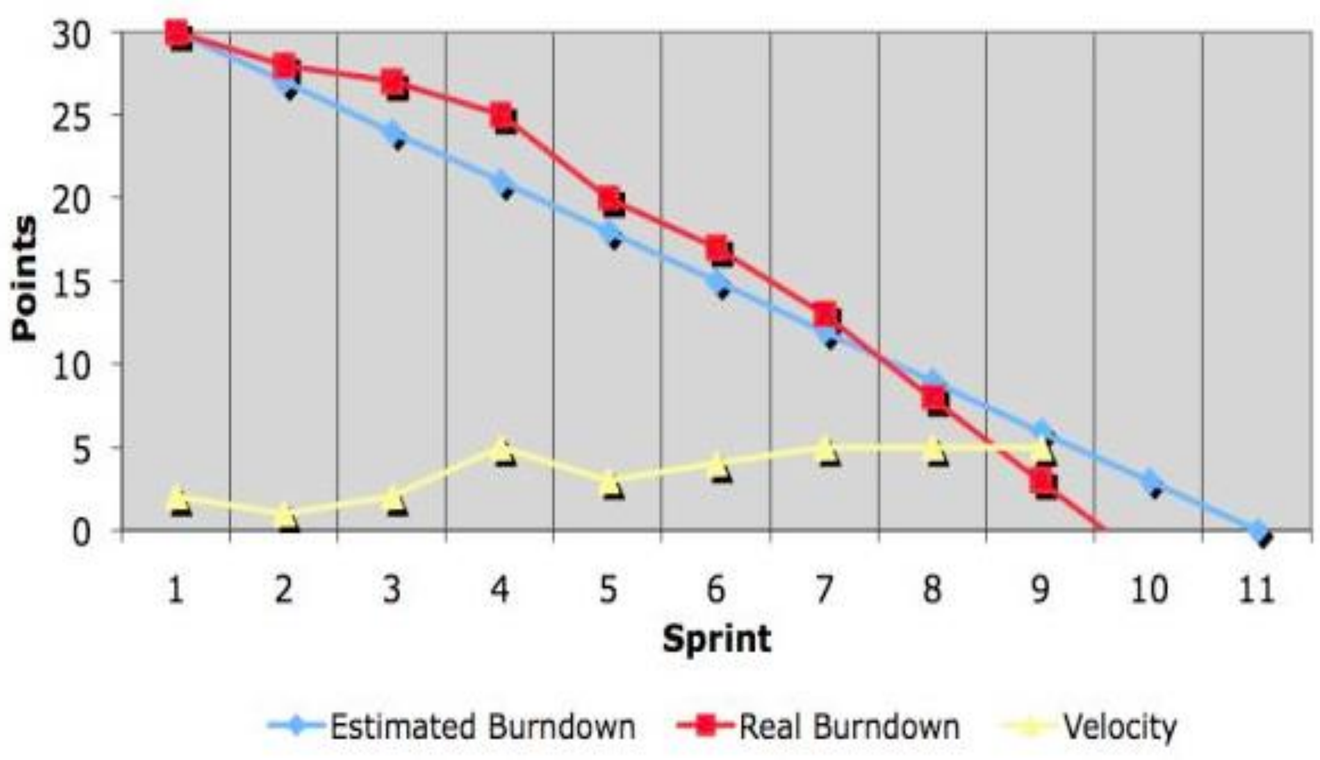
- A full task level Sprint Backlog estimated in hours is equivalent to a formal schedule (Gantt)

More work www.mountaingoatsoftware.com/agile/scrum/scrum-tools/sprint-backlog

Sprint Monitoring

Fixed Date and Time constraints

Project phase: Sprint



- Sprint Burn-down chart **monitors actual velocity**
- Scrum Master updated chart after daily standup

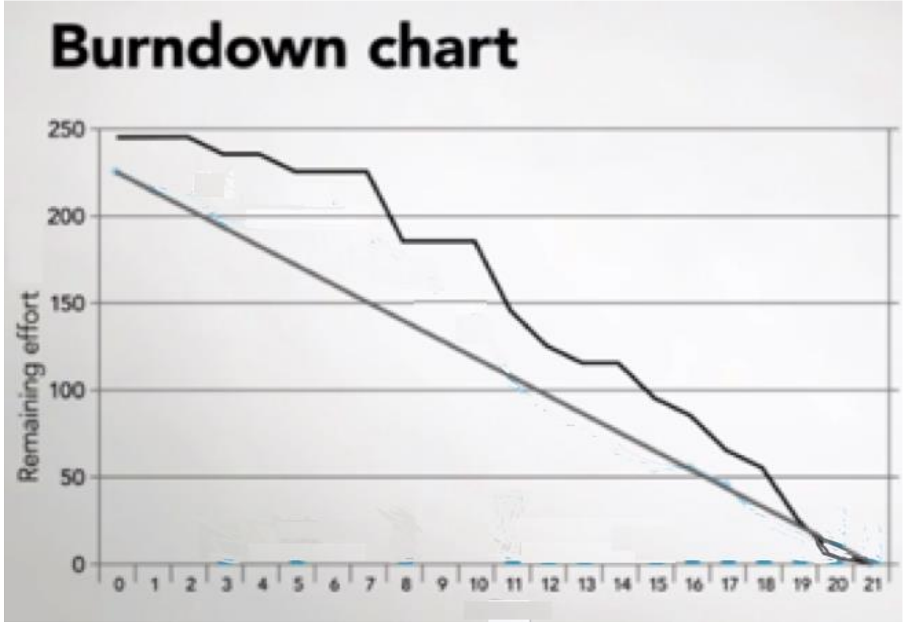
Agile Scrum Velocity

Fixed Date and Time constraints

Velocity determines the slope of the BurnDown charts

- The Scrum Master can track remaining effort
- Predict when the release milestones will be reached

Y-axis: effort



X-axis: time

- Ideal schedule is the straight line
- Actual schedule is the jagged line
- The height of the chart shows the amount of work remaining

Burn Down Chart: Activity

Product Backlog	
User Story	Story Point
Story_1	3
Story_2	5
Story_3	13
Story_4	8
Story_5	1
Story_6	3
Story_7	2

A project has this groomed **Product Backlog**, consisting of these **User Stories** which have been estimated to have these **Story Points**.

An established development team has an average ***velocity*** of **seven** User Story Points per fortnight.

1. Estimate how many weeks this team will take to deliver?

Total = sum of SP / velocity per fortnight
= 35 total SP / 7 velocity
= 5 fortnights

2. If the team actually completes the first two User Stories in two weeks, then what is the actual velocity of the team?

8 SP per fortnight, 4 per week

3. If a new User Story with Story Point=1 is added at the start of week 3, then in how many weeks do you estimate this project will take to be delivered now?

Remaining = sum of SP / velocity per fortnight
= (1+ 27) / 8
= 3.5 fortnights
= 7 weeks
New Total = 7 + 2
= 9 weeks

Burn Down Chart: Activity

A project has this groomed **Product Backlog**, consisting of these **User Stories** which have been estimated to have these **Story Points**.

An established development team has an average ***velocity*** of **seven** User Story Points per fortnight.

Product Backlog	
User Story	Story Point
Story_1	3
Story_2	5
Story_3	13
Story_4	8
Story_5	1
Story_6	3
Story_7	2

4. Will User_Story_3 fit into a single sprint?

   NO

5. What process does Scrum have for completing User_Story_3 ?

   Break it down to smaller user story(s)

Estimation Strategy Overview

Top Down strategy:

Use cost of a previous similar project, size and effort

Source Lines of Code, Function Points, Cocomo

Bottom up strategy:

Estimate individual work items and sum

WBS, Agile Story Points and Velocity

Parametric:

use project characteristics in a mathematical model

NVP, ROI, IRR

Today's aim

Become familiar with

Agile

User Stories and Story Points and Velocity

Formal

Function Point Analysis and COCOMO II

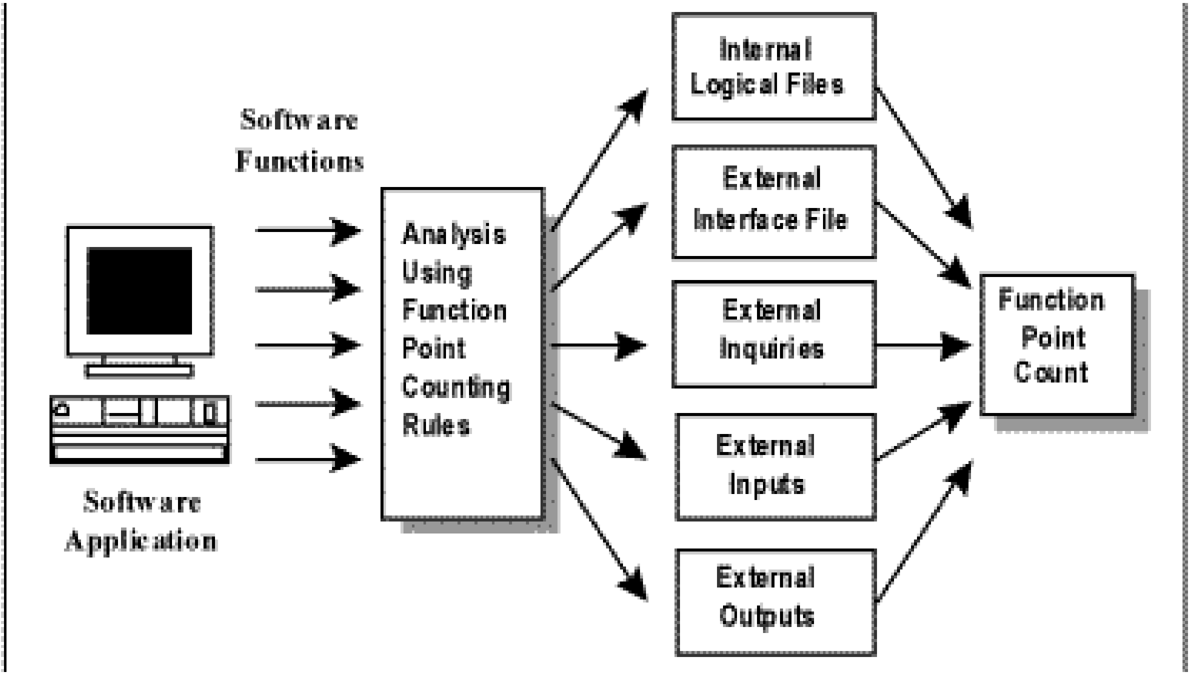
Functional Points

What are they?

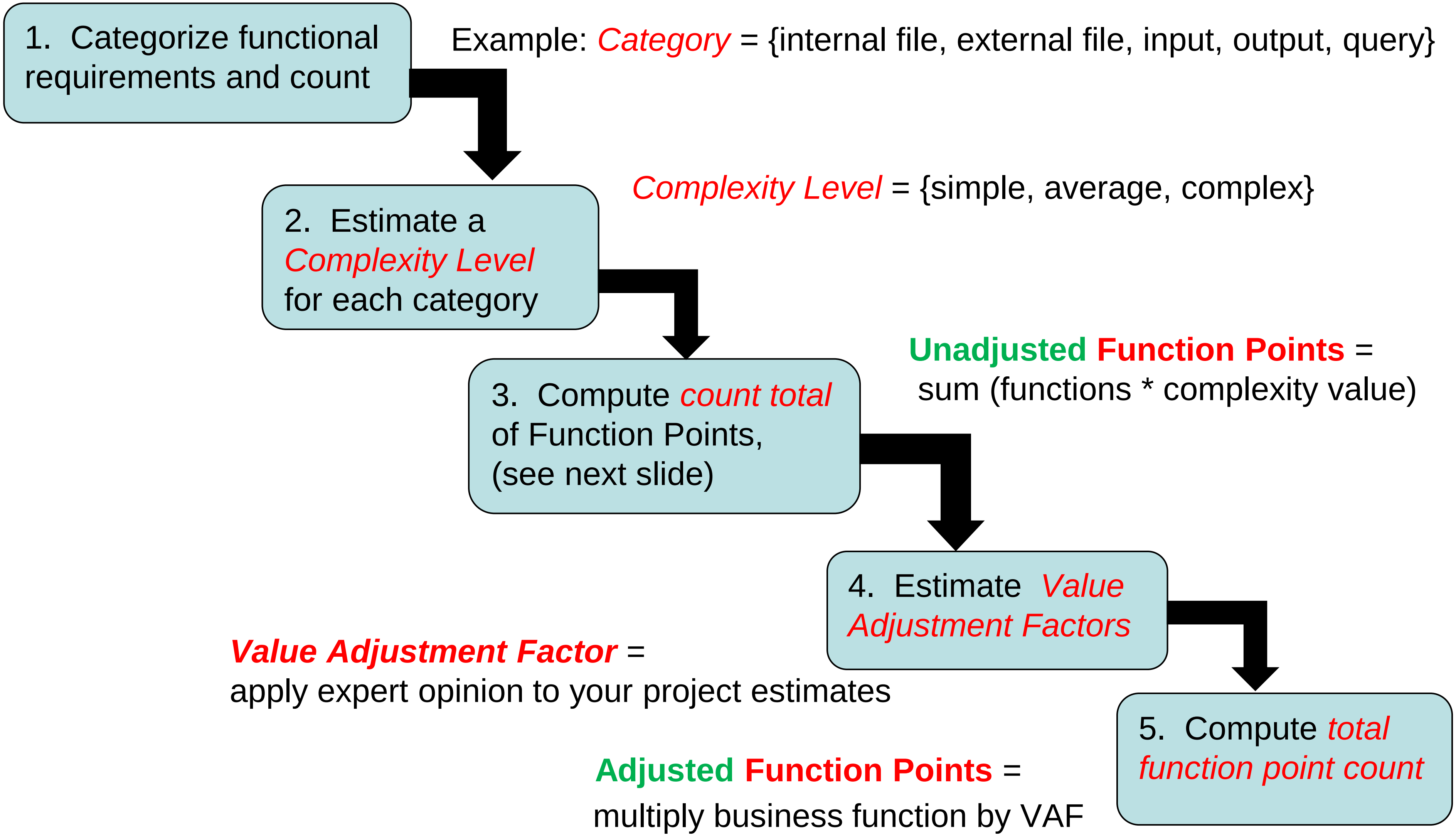
PMBOK

Historic Data

Done at any time in project lifecycle



FP Computation Steps



1. Categorize functional requirements and count

Example: *Category* = {internal file, external file, input, output, query}

2. Estimate a *Complexity Level* for each category

Complexity Level = {simple, average, complex}

3. Compute *count total* of Function Points, (see next slide)

Unadjusted Function Points = sum (functions * complexity value)

4. Estimate *Value Adjustment Factors*

Value Adjustment Factor = apply expert opinion to your project estimates

5. Compute *total function point count*

Adjusted Function Points = multiply business function by VAF

FP Computation Steps

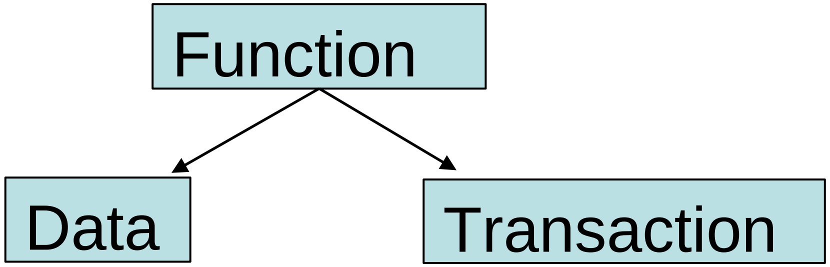
1. Categorize functional requirements and count

Example: *Category* = {internal file, external file, input, output, query}

2. Estimate a *Complexity Level* for each category

Complexity Level = {simple, average, complex}

Count functions from the Software Requirements Specification (SRS)



Data elements
Record elements

Data elements
File references

Step 2: Set Complexity Values

Historic Data

complexity values

Category	Simple Function Count	Weight	Average Function Count	Weight	Complex Function Count	Weight	Sub total
Internal Logical File	5	3		4	2	6	
External Interface File`		4		5	1	7	
External Input	2	3		4		6	
External Output	5	7	2	10	2	15	
External Inquiries/Queries	2	5		7		10	
Unadjusted Total							

Factors published from 2,192 recent Function Point projects

http://www.qsm.com/resources/function-point-languages-table

Step 3: Calculate Functional Points

Given the following business functions,
how many *Unadjusted* Function Points exist?

Fill in the table.

Category	Simple Function Count	Weight	Average Function Count	Weight	Complex Function Count	Weight	Sub total
Internal Logical File	5	3		4	2	6	
External Interface File`		4		5	1	7	
External Input	2	3		4		6	
External Output	5	7	2	10	2	15	
External Inquiries/Queries	2	5		7		10	
Unadjusted Total							

Step 3: Calculate Functional Points

Category	Simple Function Count	Weight	Average Function Count	Weight	Complex Function Count	Weight	Sub total
Internal Logical File	5	3		4	2	6	27
External Interface File`		4		5	1	7	7
External Input	2	3		4		6	6
External Output	5	7	2	10	2	15	85
External Inquiries/Queries	2	5		7		10	10
Unadjusted Total							135

Step 4: Calculate VAF

Historic Data

Give the 14 system characteristics, estimate how relevant they are to your system, use the ***typical weights***

0 = no effect
1 = incidental
2 = moderate
3 = average
4 = significant
5 = essential

Total VAF = 40

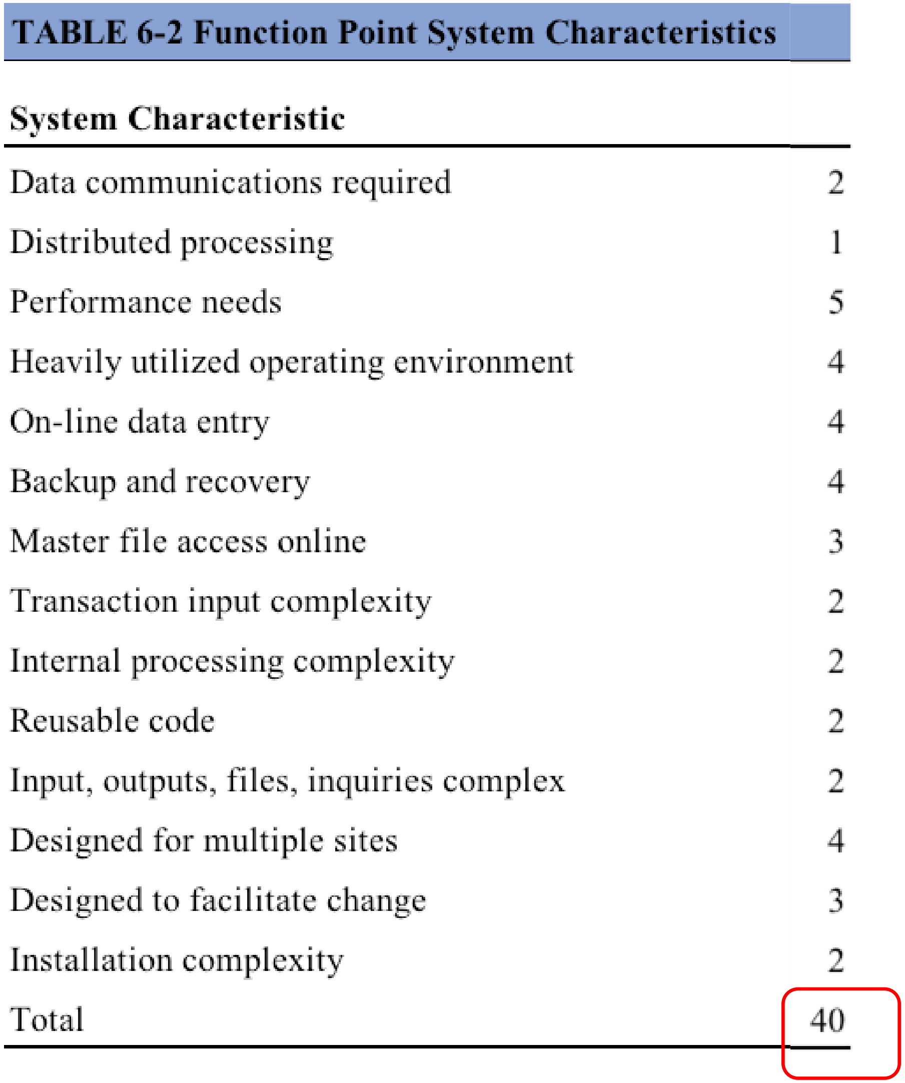


TABLE 6-2 Function Point System Characteristics

System Characteristic	
Data communications required	2
Distributed processing	1
Performance needs	5
Heavily utilized operating environment	4
On-line data entry	4
Backup and recovery	4
Master file access online	3
Transaction input complexity	2
Internal processing complexity	2
Reusable code	2
Input, outputs, files, inquiries complex	2
Designed for multiple sites	4
Designed to facilitate change	3
Installation complexity	2
Total	40

Step 5: Calculate Adjusted FP

Compute **Adjusted** Function Points using formula:

Unadjusted FP * (0.65 + 0.01 * VAF)

Project Approval Board
Don't worry! My team is hard at work coming up with an accurate cost estimation for the project.
I say we each get 3 tries and then we average the results.
J. KING MODERN ANALYST
$102,305.23
$763.21
$3,875,412.00
$738.96
$10.65
$2,153.7
$5,358.97
$27,353.66

= 135 * (0.65 + 0.01 * 40)

= 135 * (0.65 + 0.40)

= 135 * (1.05)

= 141.75 Adjusted Functional Points

COCOMO II – another strategy

The Constructive Cost Model:

Here is a playpen to try: http://csse.usc.edu/tools/cocomoii.php

Fill in the details for the Language Research Project.

Extra details to get started: let there be:

Sizing method: 135 Function Points

The Java development language

The cost per person-month is $1500

Thank You!