Faculty of Applied Science & Engineering
University of Toronto

Midterm Test

Department: Electrical & Computer Engineering
Instructor: Matt Medland
Date & Time: 3:10 p.m. – Mon., Oct. 20, 2014

Conditions: Closed Book
Duration: 50 minutes

This test counts for 20% of your final grade

Name: SOLUTIONS
(Please underline last name)

Student Number: ____________________

Question Marks

1		/20	
2		/20	
3		/20	
4		/20	
5		/20	
5		/5	[BONUS]

Total ________ /100 = ________ %

1. **[Burn-down charts, 20 marks total]**

(a) [10 marks] List a few facts that can be derived from the burn-down chart below. The solid line shows "**actual**" and the dashed line shows "**planned**"

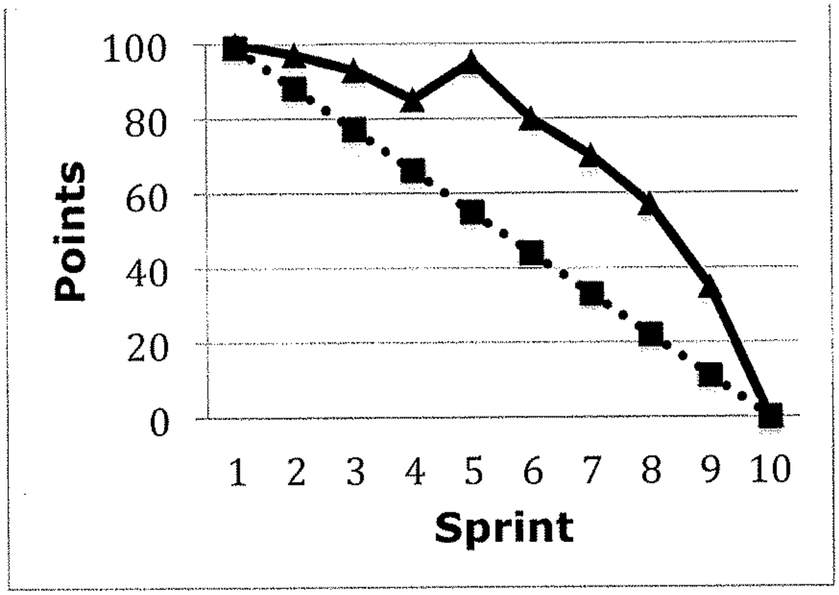

Figure 1: burn-down chart

- some days were more productive than expected, others less than expected.
- we added scope, or re-estimated upwards during sprint #5.
- slower than expected, at first, then "Superhero" push at the end
- each sprint should burn-down 10 points or so.
- other reasonable answers possible.

(b) [10 marks] List any important facts that you can **not** derive from the burn-down chart above?

- are we close to shipping anything?
- are we going to release on schedule?

2. [Planning a sprint, 20 marks total]

The following facts are given for planning the next sprint:

- features are sized in dedicated days
- a dedicated day is equal to 8 dedicated (uninterrupted) hours
- work factors convert body (or calendar) days to dedicated days
- the next sprint will be 2 weeks, or 10 workdays

(a) [10 marks] Given the team members below, how many dedicated days of feature work can be planned for the sprint?

- Ralph spends 4 dedicated hours a day writing code, and 4 hours as a team lead for the group
- Jane has 2 vacation days this sprint, and her work factor is 0.5
- Alice only works 6-hour days, but is dedicated the whole time
- Bob takes every Friday off, but works 10-hour days Monday to Thursday. His last measured work factor was 0.8

Ralph : 40 hrs / 8 = 5 days

Jane : $(10-2) \times 0.5 = 4$ days

Alice : $(6/8) \times 10 = 7.5$ days

Bob : (80 hrs / 8) × 0.8 = 8 days

TOTAL = 24.5 dedicated days

(b) [5 marks] On average, how many effective developers do you have per day during the sprint?

$24.5/10 = 2.45$ dedicated developers per day

(b) [5 marks] How do your calculations above relate to the capacity constraint? i.e. give the values of **F**, **N** and **T** in the formula:

$$\mathbf{F \le N \times T}$$

$F \le 24.5$

$N = 2.45$

$T = 10$

3. [Crossing the Chasm, 20 marks total]

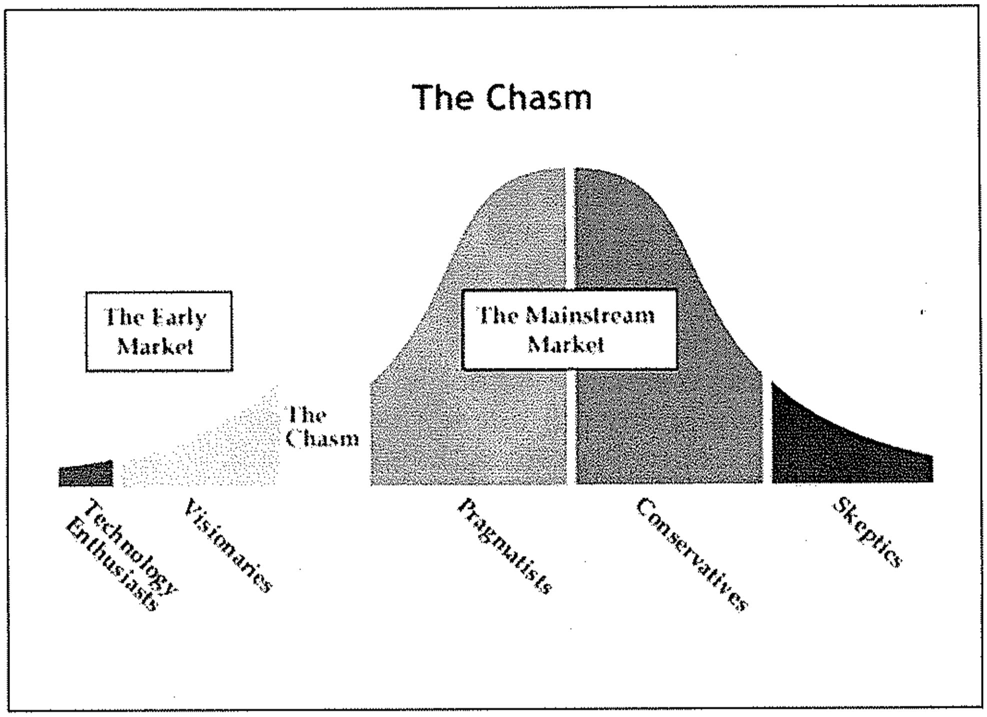

Figure 2: The Chasm

(a) [5 marks] List a couple of defining characteristics of the early market (left of chasm) and the mainstream market (right of chasm)

- [MUST SAY] early is tolerant of bugs & unpolished features. Late expect everything to just work
- early are experts, late are not.

(b) [15 marks] Why do many software companies fall into this metaphorical chasm, what happens to them when they get there, and what could have kept them out?

- baggy software, bad design, can't go from prototype to product, don't know how to market or sell to masses, etc.
- companies die in the chasm!
- proper planning keeps them out.

4. [Dealing with overage, 20 marks]
You are the manager of a software development department, and you've just figured out that your next release is not going to be on time. What are some things you can do to remedy the situation?

- [MUST SAY] cut features, decrease scope
- [MUST SAY] extend dates
- [MUST SAY] combination of above two
- improve work-factors (less distraction)
- reassign features among developers to take advantage of expertise
- postpone/cancel vacations
- hire more developers

5. [Important planning questions, 20 marks total]
(a) [10 marks] What are three (3) important questions a software company must answer when planning a software development effort?

[Hint: first question is ① "what are we building?"]

② by when will it be ready?

③ with how many developers?

(b) [5 marks] What is the more important question that many companies forget to ask?

can we do all 3 above at once?

(c) [5 marks] At what points in the software development effort do we need to take into account the answers to these questions? (Circle the answers that apply)

[✓ **before**]	[✓ **during**]	[✓ **after**]
don't start if it's not "balanced"	always keep plan "balanced"	to calculate work-factors for next release

6. [Apache server, 5 marks – BONUS]
Apache server modules can be compiled statically into the **httpd** binary by passing options to the **configure** script, or as dynamic shared objects (DSOs) using **apxs**. How do you tell the Apache server to load the DSO modules at runtime?

LoadModule directive in httpd.conf

Briefly, what is the difference between statically compiling a module and runtime loading a DSO module?

when statically compiled with options to configure script the code implementing the module's functionality is built into the httpd executable program.

if built as a DSO, the module's code exists in a shared object (.so, .dylib, .dll – depending on the OS platform) and is loaded by httpd as needed at runtime.