Syllabus
Assets/**Q-Kinds**

Read ~ 35 pages: Chs 1, 2-2.4 from 9e – or Chs 1,2,3 from 8e.

SWE Introduction

[nox]
Pgm Sizes – [non-std – old-school typewritten "page" = 50-55 lines/page & 12 5-letter words/line]
XXS = 25 LOC, half page (avg typical algorithm size)
XS = 100 LOC, 2 pgs (AKA Tiny)
SM = 500 LOC, 10 pgs
MD = 2,500 LOC, 50 pgs
LG = 10,000 LOC, 200 pgs
XL = 50,000 LOC, 1,000 pgs
XXL = 250,000 LOC, 5,000 pgs
XXXL = 1,000,000+ LOC, 20,000 pgs

Pgmg Lang power == fewer LOC to get the job done (in an easily understandable way)

Why we need SWE?

[Jones][Chaos]

o- 20% **to 80%** of all non-small (> 20 pgs) projects **Fail** – **% depends on who you talk to**
o-- or experience a **massive overrun** (> 33%) in cost/effort or timeline, usually both
→ paid for by grudging customer (gov't agency), who probably won't buy from you again
o-- **We will go over** these reports & some exciting massive failures in lecture

o- **2 Kinds of Failure**: 1) Never delivered, and 2) Delivered but deemed **unsuitable** to use
o-- **Unsuitable** – 1) too slow, 2) too complicated to operate,
3) too many errors/inaccuracies, and/or 4) breaks down too often (low MTBF)

Why do projects Fail? **4 major reasons**:

o- **Complexity** (scales exponentially ceteris paribus)

o- **Mgrs** (mostly lack of knowledge)
o-- Fail to manage devr (developer) **morale** → much lower productivity == much more effort
o-- Fail to arrange adequate/effective **comm w users**
o-- Fail to see **looming risk** events in time
o-- Fail to **gel group** into a team, or maintain the team

o- **Comm poor** with "users" (built ill, unsuitable product)
o-- (Extra hard) Not every project has an individually identified set of users

o- **Bad Prediction** of "The Plan" (Features/Effort/Timeline – AKA the "Project Triangle")

SWE Status (eg, today, how does SWE "look")

The Good
o- Pgmr Need is still growing faster than all other occupations

The Bad
o- "SWE" coined 1968 (at NATO conf), after a decade-plus of failed big-gov projects
o- 20% to 80% of all **M-L-XL+** SWE projects fail today, [Standish/Chaos][Jones]
o- SW Defects (= shipped bugs) cost US alone ~$60B [Tassey 2002] – likely an under-estimate

The Ugly
(*)** No one knows how to build SW (>= medium size, 50pgs) reliably
o- Parnas/Lawford: 30+ years SWE research hasn't helped [Parnas 2003]
(*)* Key: "Quality S/W" == Bug-Free – never achieved, even for the "good" products
o-- 10s of $$M spent on SWE **research** to date (**pbb >> $1B**)
o-- NB, Dave Parnas buddy w Fred Brooks (IBM & taught at UNC Chapel Hill, good CS dept)

o- **"No Silver Bullet", Brooks**, 1985 -- still true today – **we will read** this paper
o- SWE taught for BSCS for >40 years

o- Numerous **SWE "Quality" Groups** now exist, some >30 years old
o-- SWE today is **barely** better understood than 4+ decades ago
– the "barely" award goes to the **Agile Manifesto** crowd – **we will read** & memorize it

o- Sample of major SWE Quality Groups
ABET -- for Univ CS Dept curriculum accreditation – undergraduate degrees
ACM -- for model CS curriculum – undergraduate degrees
SEI -- **CMMI** pseudo-metrics + a large pile of docs on how to do SWE
Computer.org – Certs (Certifications)
SWEBOK -- a mishmash, some useful (SWE Body of Kno.wledge) [Bourque 2014)
SWECOM -- ditto (SWE Competency Model)
ISO 9xxx group (Euro) – SWE standards docs

Assets/**Mini-SWE** Prelims & Rules

Pressman & Maxim: 8th ed vs 9th ed – line by line correspondence of section headings

Ch 1. The Nature of SW 1 1 . 1 The Nature of Software 3 1. 1. 1 Defining Software 4 1. 1.2 Software Application Domains 6 1. 1.3 Legacy Software 7 2. 1 Defining the Discipline 15 2. 2 The Software Process 16 2.2.1 The Process Framework 17 2. 2.2 Umbrella Activities 18 2. 2.3 Process Adaptation 18 2.3 Software Engineering Practice 19 2.3.1 The Essence of Practice 19 2.3.2 General Principles 2 1 2.5 How It All Starts 26	Ch 1 Software And Software Engineering 1 1.1 The Nature of Software 4 1.1.1 Defining Software 5 1.1.2 Software Application Domains 7 1.1.3 Legacy Software 8 1.2 Defining the Discipline 8 1.3 The Software Process 9 1.3.1 The Process Framework 10 1.3.2 Umbrella Activities 11 1.3.3 Process Adaptation 11 1.4 Software Engineering Practice 12 1.4.1 The Essence of Practice 12 1.4.2 General Principles 14 1.5 How It All Starts 15
3. SW Process Structure + 4. Process Models 3. 1 A Generic Process Model 31 3.2 Defining a Framework Activity 32 3.3 Identifying a Task Set 34 3.5 Process Assessment and Improvement 37 4.l Prescriptive Process Models 41 4.1.1 The Waterfall Model 41 4.1.2 Incremental Process Models 43 4.1.3 Evolutionary Process Models 45 4.3 The Unified Process 55 4.6 Product and Process 62	Ch 2 Process Models 20 2.1 A Generic Process Model 21 2.2 Defining a Framework Activity 23 2.3 Identifying a Task Set 23 2.4 Process Assessment and Improvement 24 2.5 Prescriptive Process Models 25 2.5.1 The Waterfall Model 25 2.5.2 Prototyping Process Model 26 2.5.3 Evolutionary Process Model 29 2.5.4 Unified Process Model 31 2.6 Product and Process 33