Software Engineering CS3003 (2021-2022)

Lecture 1: Introduction

(Steve Counsell –
steve.counsell@brunel.ac.uk)

About me

- Joined Brunel in 2004
- Joint Head of “BSEL” with Nour
  - Brunel Software Engineering Lab
- Taught this module since 2019
- Do research in a lot of topics included in the schedule:
  - Industry involved
  - BT, Sky, Bloomberg, Ericsson and SMEs
- Prefer ‘Steve’

Structure of this lecture

- The module team
- Overview of the module
  - Learning outcomes, assessment, programme of study
- Introduction to Software Engineering
- Reading for the week
- Note: these slides will be put on BBL straight afterwards

Module team

- Module Leader and lecturer: me!
- Other lecturer: Dr Giuseppe Destefanis giuseppe.destefanis@brunel.ac.uk
  - 2 lectures
- GTAs (will attend labs)

The module structure

Lectures (Steve, Giuseppe):

- Provide an understanding of the area
- All lectures are on Wednesday 11.00-13.00
  - Live streamed
  - Break half-way
  - Lecture slides will be put up on BBL every Monday morning for the week
- Course Texts
  - Plenty of background reading will be provided
  - Why do background reading?

Module structure (cont.)

Labs (Steve, Giuseppe & GTAs):

- Extend your knowledge of relevant topics
- The lab sheets can be done in your own time
  - They require no specialist software
  - No requirement for you to attend labs physically
- All labs are on Wednesday 10am-11am
  - Lab sheets put up on BBL Monday morning
- My thoughts on the lab sheets will be posted after each lab session

Lecture schedule

Week	Lecture Topic	Lecturer	Week Commencing
1	Introducing the module and Software Engineering	Steve Counsell	20th Sept.
2	Software maintenance and Evolution	Steve Counsell	27th Sept.
3	Software metrics	Steve Counsell	4th Oct.
4	Software structure, refactoring and code smells	Steve Counsell	11th Oct.
5	Test-driven development	Giuseppe Destefanis	18th Oct.
6	Software complexity **Coursework released Tues 26th Oct.**	Steve Counsell	25th Oct.
7	**ASK week**	**N/A**	**1st Nov**
8	Software fault-proneness	Steve Counsell	8th Nov.
9	Clean code	Steve Counsell	15th Nov.
10	Human factors in software engineering	Giuseppe Destefanis	22nd Nov.
11	SE techniques applied in action	Steve Counsell	29th Dec.
12	Guest Lecture (tba) **Coursework hand-in 6th December**	Guest Lecture	6th Dec.

Please use this lecture schedule

- The previous slide is in the study guide
- Please use this schedule as the guide to lectures this term and no other

Lab schedule

Week	Labs	Week Commencing
1	**No labs**	20th Sept.
2	Lab (Introduction)	27th Sept.
3	Lab	4th Oct.
4	Lab	11th Oct.
5	Lab	18th Oct.
6	**No lab**	25th Oct.
7	**ASK week**	**1st Nov.**
8	Lab	8th Nov.
9	Catch-up Lab	15th Nov.
10	Work on coursework (no Lab)	22nd Nov.
11	Work on coursework (no Lab)	29th Nov.
12	**No lab**	6th Dec.

Please use this lab schedule

- The previous slide is in the study guide
- Please use this schedule as the guide to labs this term and no other

Assessment

Coursework

- Pass/fail (threshold)
- Released week **6** handed in week **12**
- Second attempt if do not pass first time

Assessment (cont.)

Exam

- Determines grade for those who pass coursework
- 5 “essay-style” questions on the paper
  - Answer any 4 questions of you choice
  - Format of questions in parts
    - Example

Benefits of structure include:

- Less risk/stress for finals

Learning outcomes

LO1: Describe the **attributes** of **quality software** and the **implications** of poorly designed software

LO2: Describe and **evaluate** the processes and techniques which may be used to produce quality software and be able to create software artefacts which display these attributes

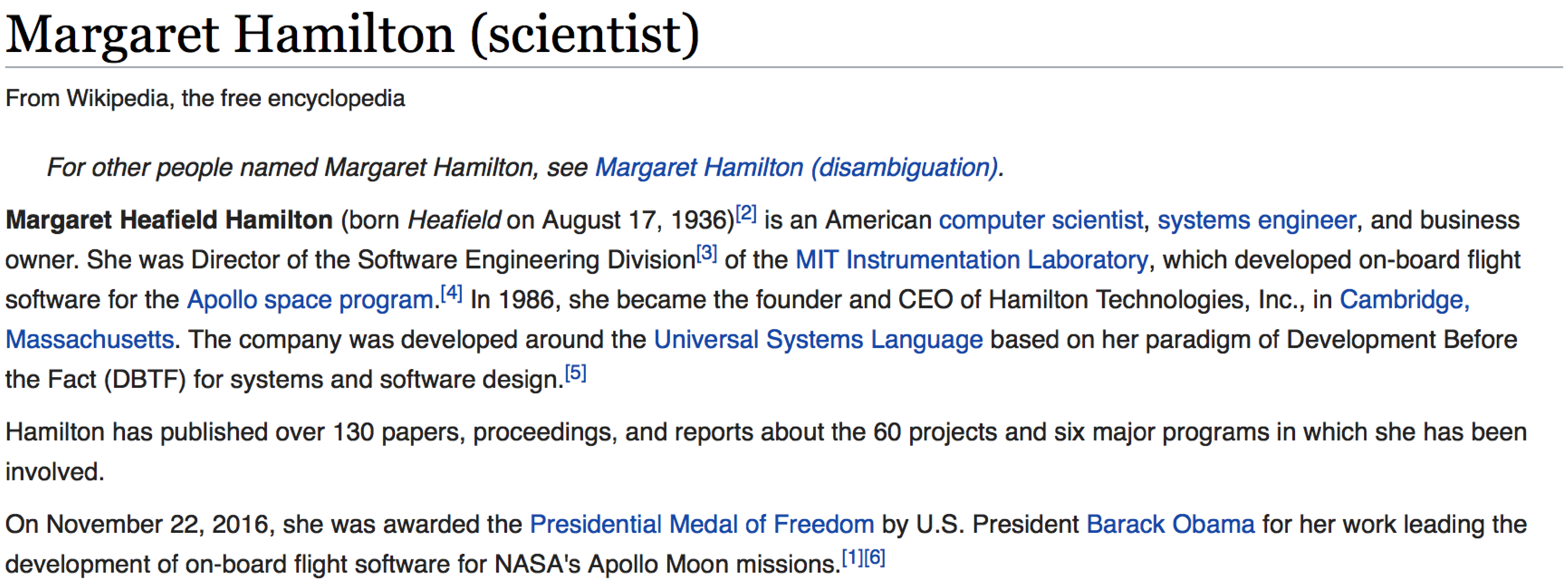
LO3: Critically evaluate, select and appraise software **metrics** in order to assess software process and product attributes

A few other things

- There is no Coderunner in this module
- I won’t be expecting you to write code, but I do expect you to follow some relatively simple Java and pseudo code
- Please remember to read your emails
  - I will replicate ***all*** information I send to you by email on the CS3003 home page of BBL

Brief Introduction to Software Engineering

MT0A
CR1A
MT1A
MT2A
CR1A
CP1A
CP1A
TY1A
CR1A
LP1A
MT2A



Margaret Hamilton (scientist)

From Wikipedia, the free encyclopedia

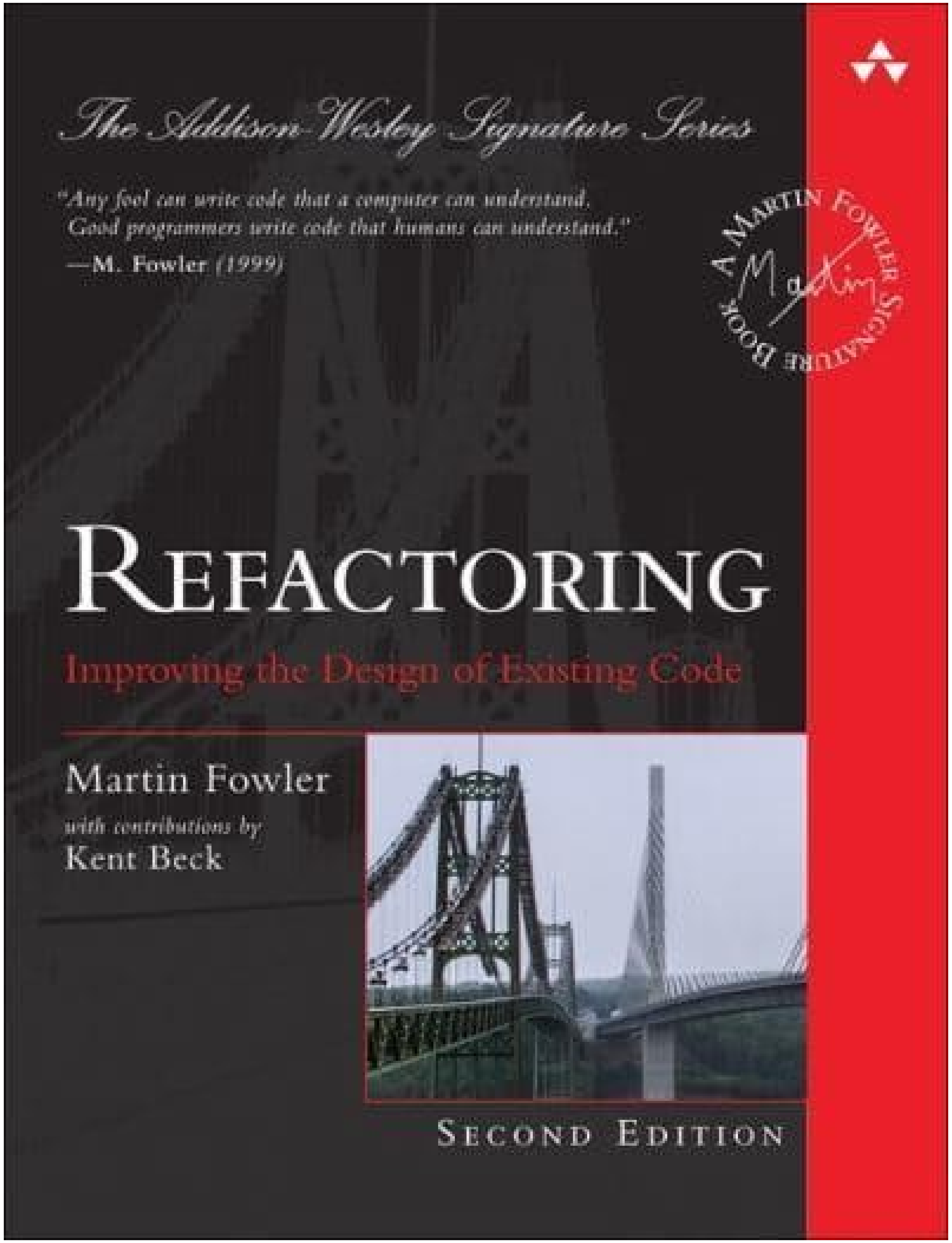
For other people named Margaret Hamilton, see Margaret Hamilton (disambiguation).

Margaret Heafield Hamilton (born *Heafield* on August 17, 1936)[2] is an American computer scientist, systems engineer, and business owner. She was Director of the Software Engineering Division[3] of the MIT Instrumentation Laboratory, which developed on-board flight software for the Apollo space program.[4] In 1986, she became the founder and CEO of Hamilton Technologies, Inc., in Cambridge, Massachusetts. The company was developed around the Universal Systems Language based on her paradigm of Development Before the Fact (DBTF) for systems and software design.[5]

Hamilton has published over 130 papers, proceedings, and reports about the 60 projects and six major programs in which she has been involved.

On November 22, 2016, she was awarded the Presidential Medal of Freedom by U.S. President Barack Obama for her work leading the development of on-board flight software for NASA's Apollo Moon missions.[1][6]

Some disciplines are long established



The Addison-Wesley Signature Series

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."
—M. Fowler *(1999)*

A Martin Fowler Signature Book

Refactoring

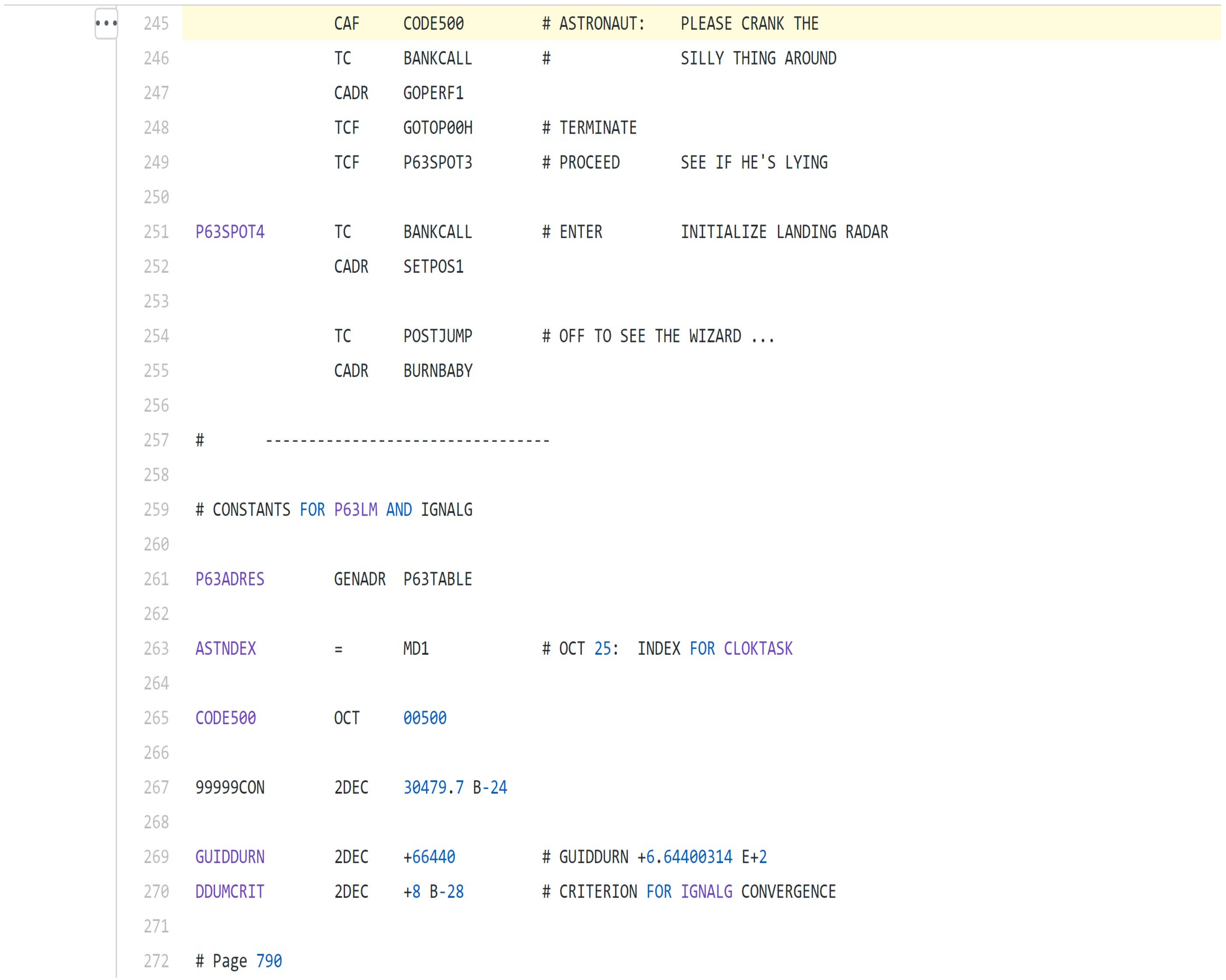
Improving the Design of Existing Code

Martin Fowler

with contributions by
Kent Beck

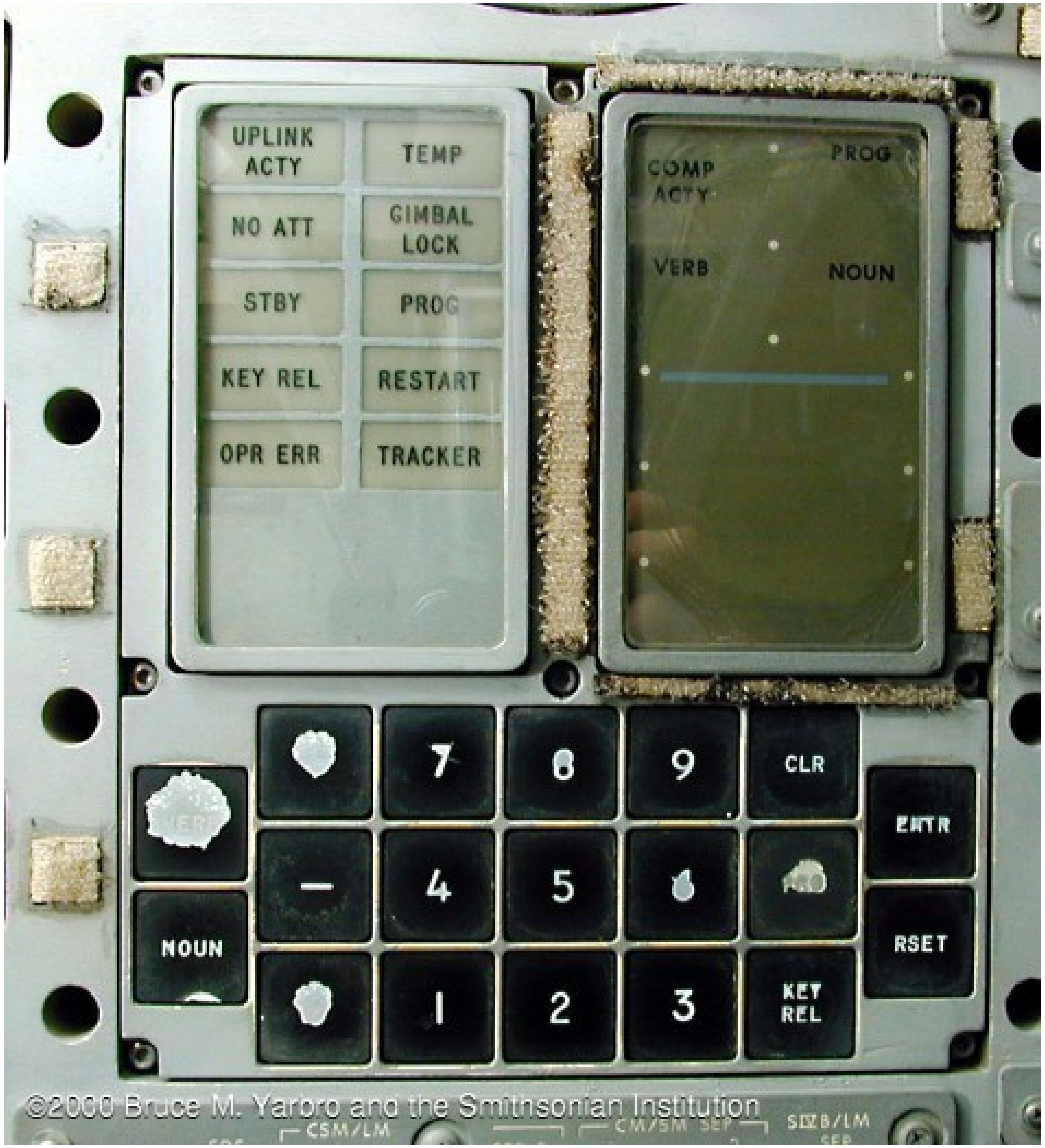
Second Edition

But not yet perfect

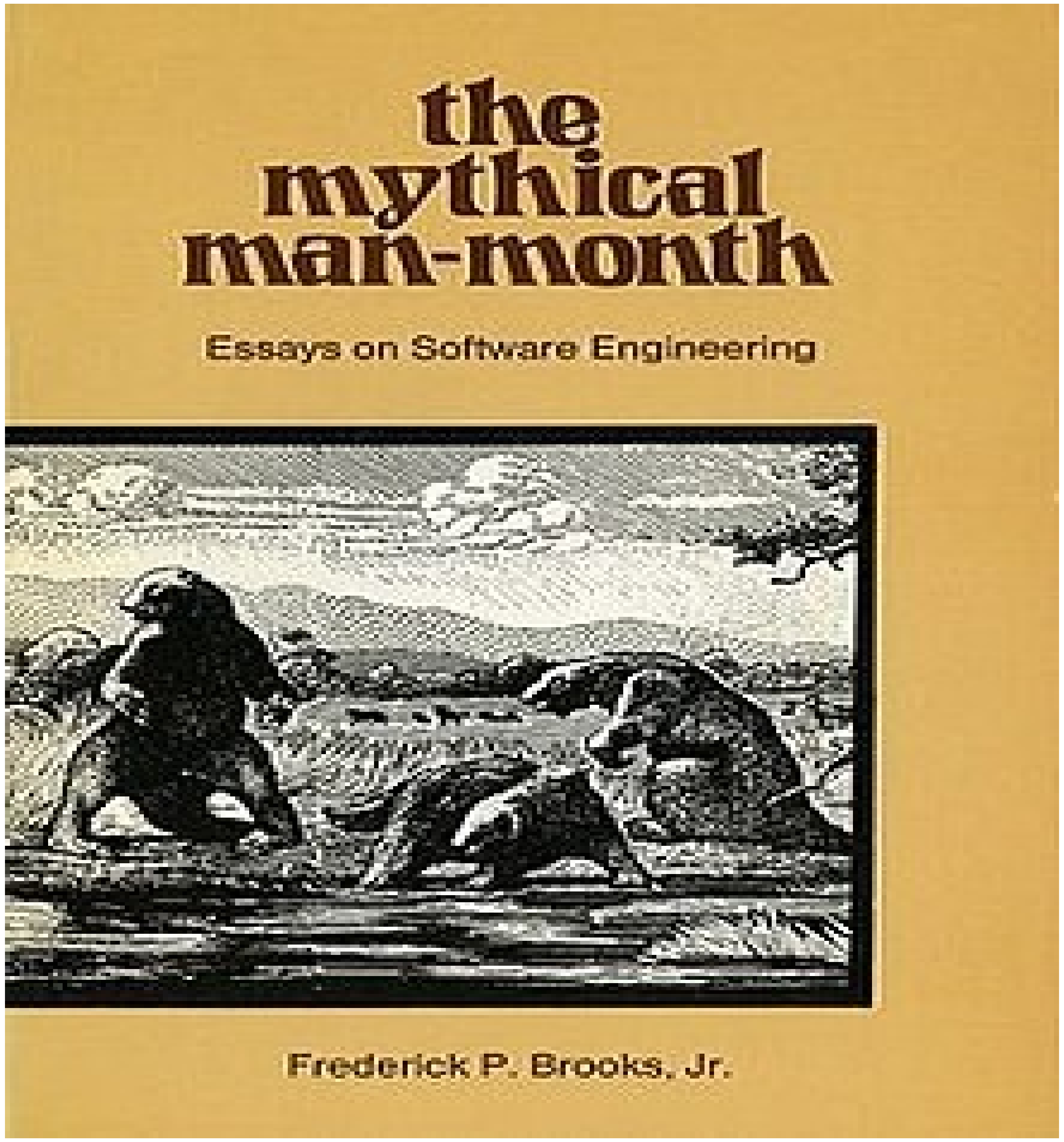


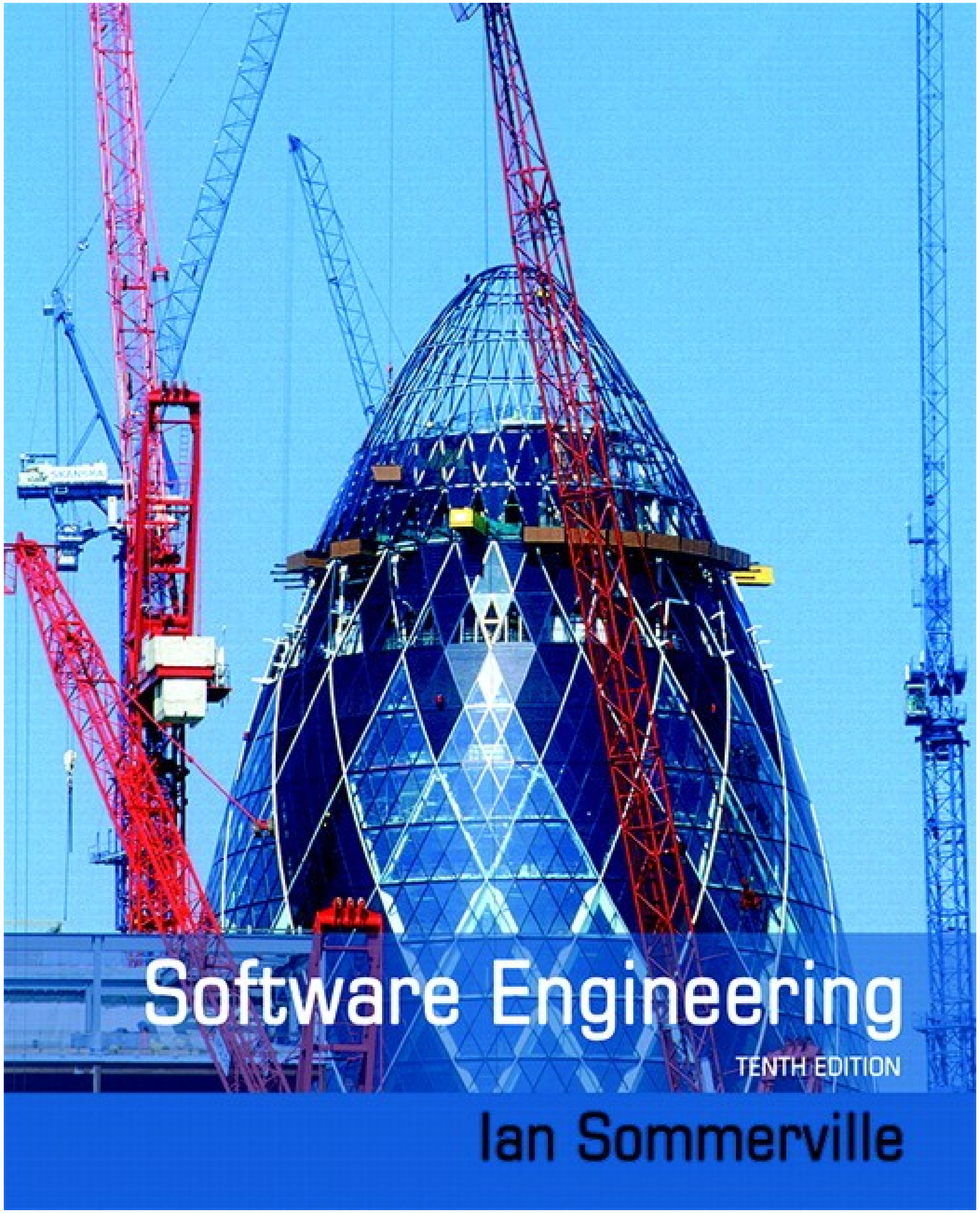
```
                CAF     CODE500         # ASTRONAUT:     PLEASE CRANK THE
                TC      BANKCALL        #                SILLY THING AROUND
                CADR    GOPERF1
                TCF     GOTOP00H        # TERMINATE
                TCF     P63SPOT3        # PROCEED        SEE IF HE'S LYING

P63SPOT4        TC      BANKCALL        # ENTER          INITIALIZE LANDING RADAR
                CADR    SETPOS1

                TC      POSTJUMP        # OFF TO SEE THE WIZARD ...
                CADR    BURNBABY

#       ---------------------------------

# CONSTANTS FOR P63LM AND IGNALG

P63ADRES        GENADR  P63TABLE

ASTNDEX         =       MD1             # OCT 25:   INDEX FOR CLOKTASK

CODE500         OCT     00500

99999CON        2DEC    30479.7 B-24

GUIDDURN        2DEC    +66440          # GUIDDURN +6.64400314 E+2
DDUMCRIT        2DEC    +8 B-28         # CRITERION FOR IGNALG CONVERGENCE

# Page 790
```

Your phone


UPLINK ACTY
TEMP
NO ATT
GIMBAL LOCK
STBY
PROG
KEY REL
RESTART
OPR ERR
TRACKER
COMP ACTY
PROG
VERB
NOUN
7
8
9
CLR
ENTR
—
4
5
6
RSET
NOUN
1
2
3
KEY REL
CSM/LM
CM/SM SEP
SIVB/LM SEP

A great book


the mythical man-month
Essays on Software Engineering
Frederick P. Brooks, Jr.


Software Engineering
TENTH EDITION
Ian Sommerville

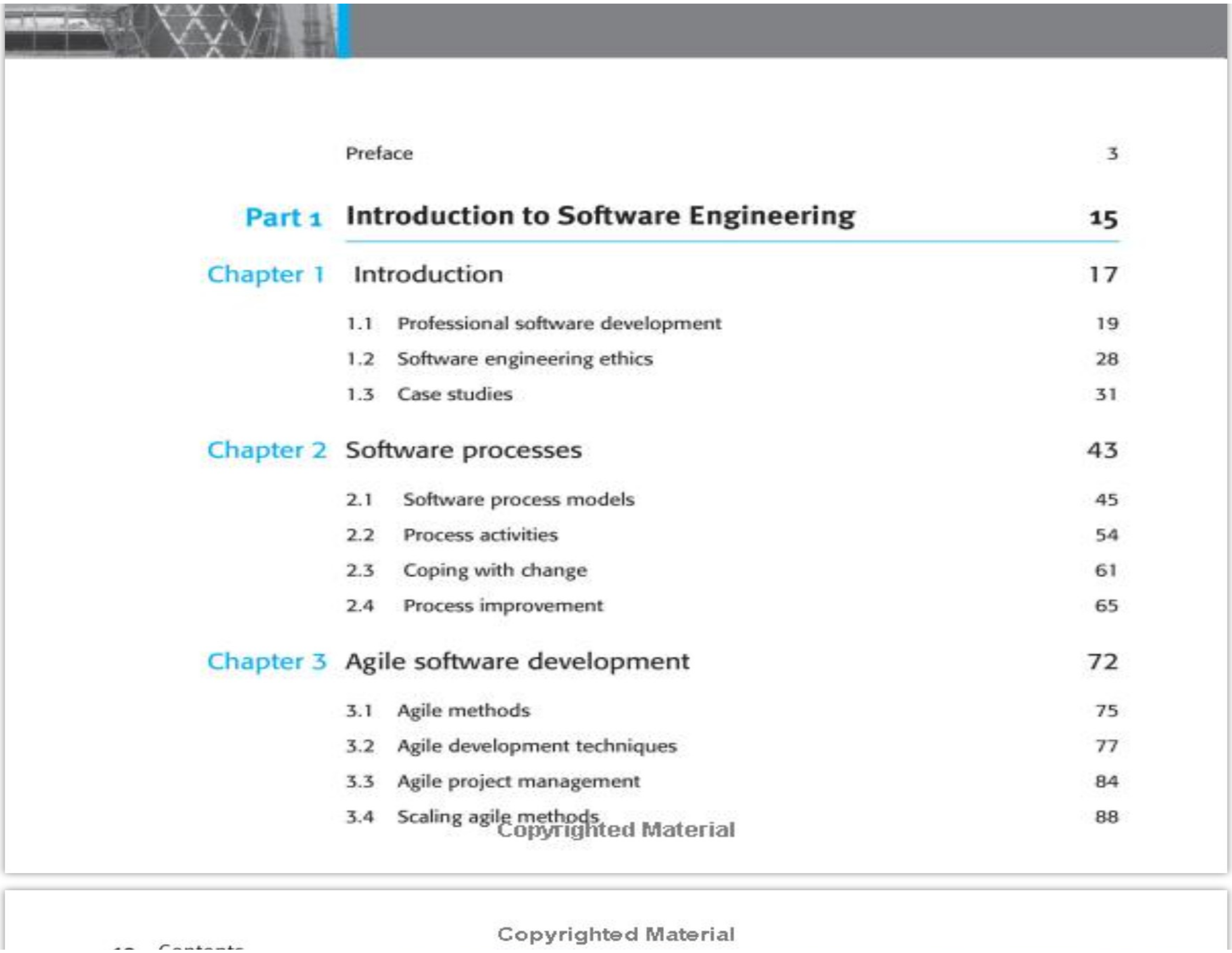
Checklist - recap

- On every Monday morning before midday on BBL:
  - Lecture slides in the “Lectures” folder
  - A lab sheet Word document
    - In the “Lab sheets and lab materials” folder
    - No need to attend lab
    - Lab thoughts of mine will be posted straight after the lab

- Reading:

-Chapter 1 of Somerville
- https://software.nasa.gov/ (grab some of their code/tools)
- listen to the AGC interface of Apollo 11:
https://www.youtube.com/watch?v=hyhI85Rd1kI

Table of contents



- Thanks for listening!
- Have a good rest of week