上海科技大学
ShanghaiTech University

CS132: Software Engineering

Zhihao Jiang

SIST@ShanghaiTech

Lecture 1: Introduction

What is software engineering?

Science vs. Engineering

Science vs. Engineering

- Science: Theoretically how something can be achieved
  - How to generate Ammonia (NH_3)?
  - $N_2 + 3H_2 \leftrightarrow 2NH_3 \uparrow$

- Engineering: How to achieve the goal efficiently and economically with existing constraints
  - In 1774: $2NH_4Cl + Ca(OH)_2 \leftrightarrow CaCl_2 + 2NH_3 \uparrow + 2H_2O$
  - In 1898: $CaC_2 + N_2 \rightarrow CaCN_2 + C \text{ and } CaCN_2 + 3H_2O \rightarrow CaCO_3 + 2NH_3 \uparrow$
  - In 1908: $N_2 + 3H_2 \leftrightarrow 2NH_3 \uparrow$ under high pressure with catalysts

Software “Science”

- Algorithm
- Theory of computation
- What problems can be solved computationally?
- How can we express the solution rigorously without ambiguities?
  - Formal languages
- Can we solve problems efficiently?
  - Computational complexity theory

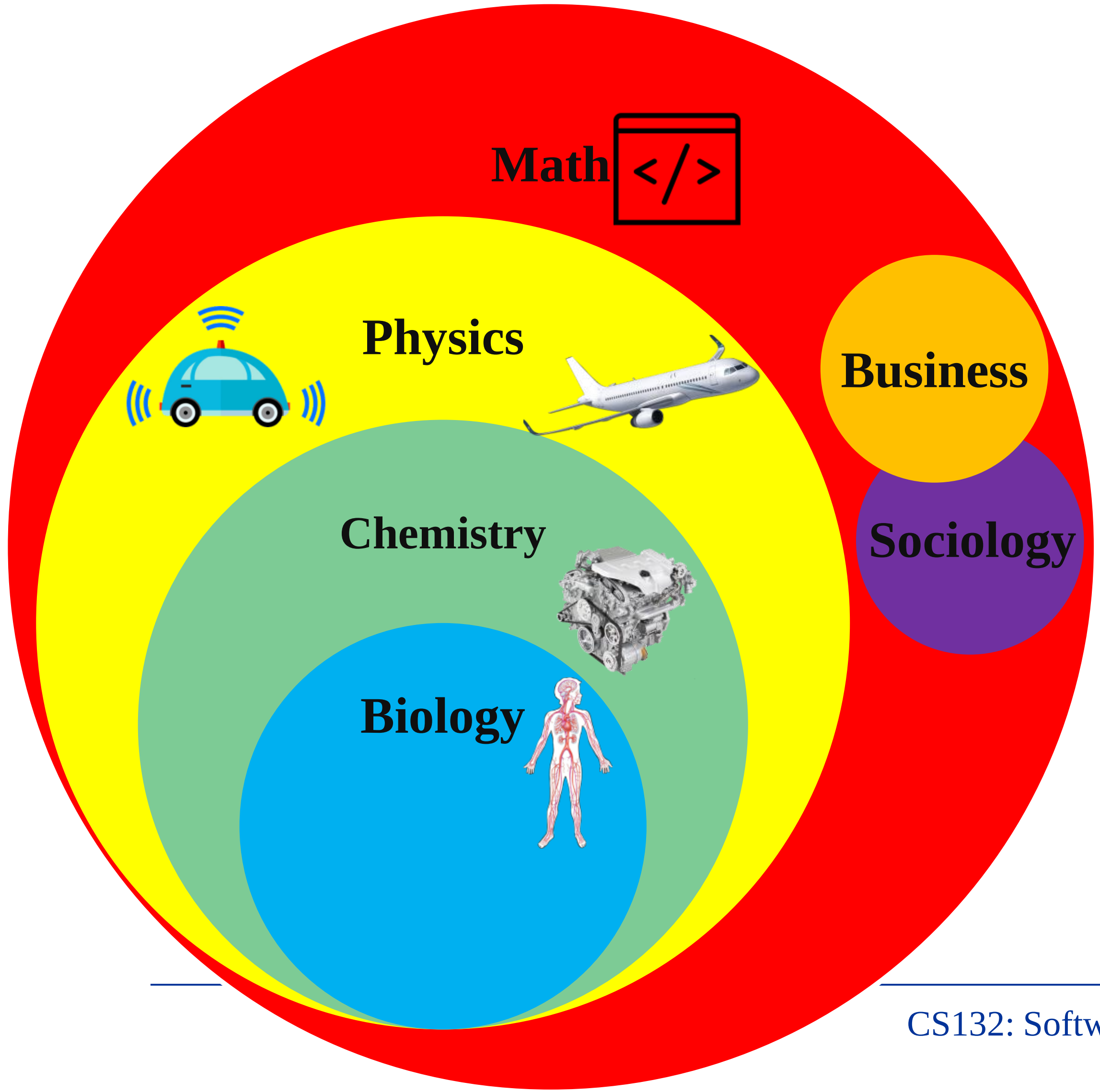
Software Engineering: Definition

- "The establishment and use of sound engineering principles in order to obtain economical software that is reliable and works efficiently on real machines …" [Fritz Bauer, at the 1st NATO Conference on Software Engineering, 1969]

“The Software Crisis”

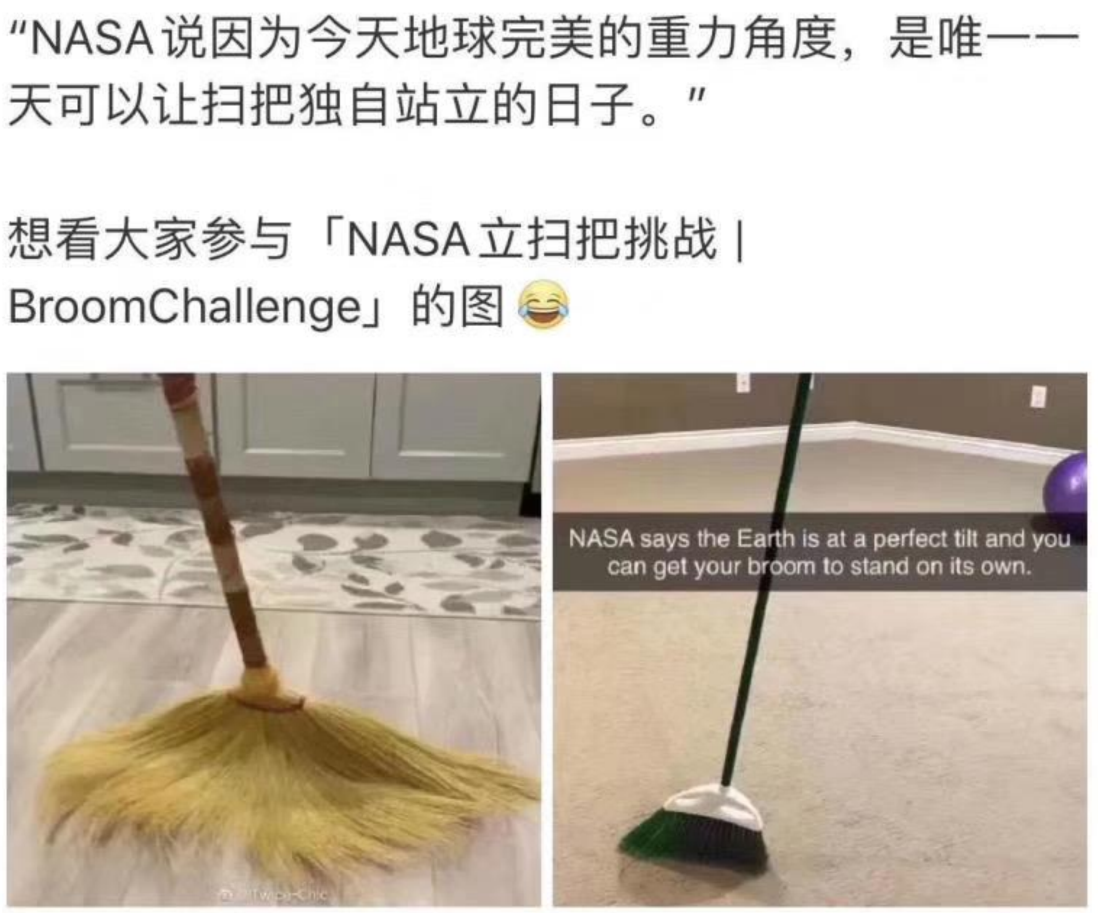
- The need to develop larger, more diverse software systems
- 28% of software projects are “success”
  - 51% seriously late, over budget and lacking expected features
  - 18% cancelled outright
- Increasing complexity
  - Use to be one single task on a specific computer
  - More functionalities
  - In more conditions
  - On more platforms
  - By more people

An Inter-disciplinary Field



- Software is in math domain
  - Amazon negative quantity bug
- The domain specific constraints have to be encoded or considered in the software
- The domains with more constraints are also less understood

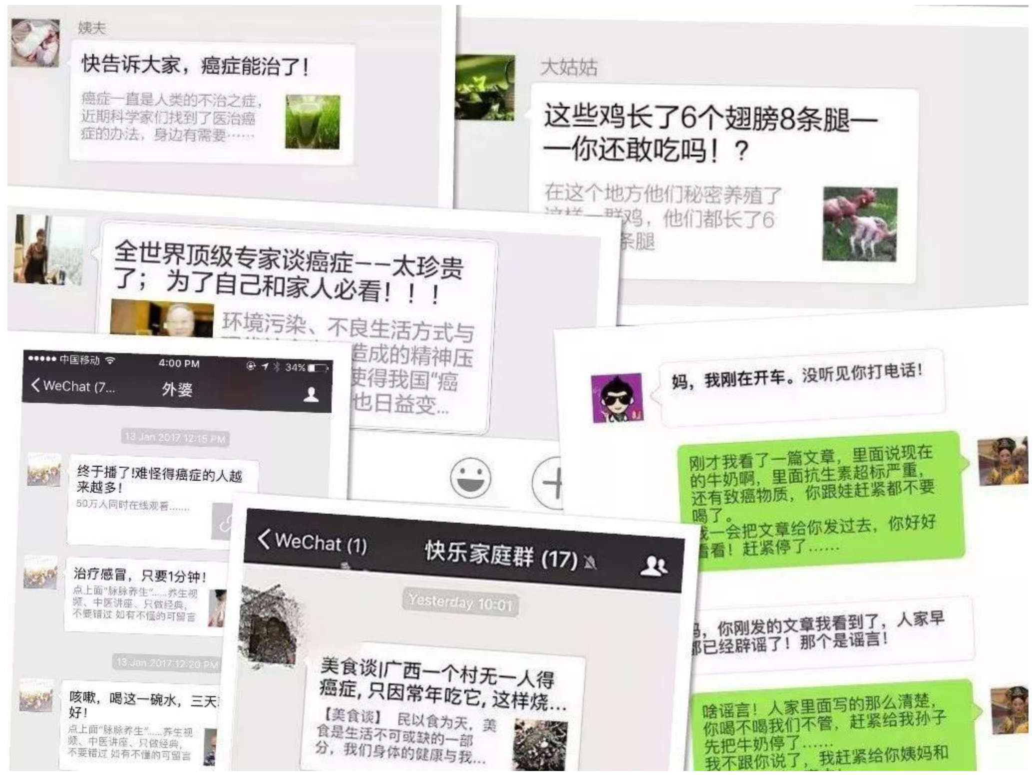
Which fake news is easier to tell?



“NASA说因为今天地球完美的重力角度，是唯一一天可以让扫把独自站立的日子。”

想看大家参与「NASA立扫把挑战 | BroomChallenge」的图 😂

NASA says the Earth is at a perfect tilt and you can get your broom to stand on its own.



姨夫

快告诉大家，癌症能治了！

癌症一直是人类的不治之症，近期科学家们找到了医治癌症的办法，身边有需要……

大姑姑

这些鸡长了6个翅膀8条腿——你还敢吃吗！？

在这个地方他们秘密养殖了这样一群鸡，他们都长了6条腿

全世界顶级专家谈癌症——太珍贵了；为了自己和家人必看！！！

环境污染、不良生活方式与造成的精神压使得我国“癌也日益变...

●●●●● 中国移动 4:00 PM 34%

WeChat (7... 外婆

13 Jan 2017 12:15 PM

终于播了!难怪得癌症的人越来越多!

50万人同时在线观看.......

治疗感冒，只要1分钟!

点上面“脉脉养生”......养生视频、中医讲座、只做经典，不要错过 如有不懂的可留言

13 Jan 2017 12:20 PM

咳嗽，喝这一碗水，三天好!

点上面“脉脉养生”......养生视频、中医讲座、只做经典，不要错过 如有不懂的可留言

WeChat (1) 快乐家庭群 (17)

Yesterday 10:01

美食谈|广西一个村无一人得癌症，只因常年吃它，这样烧...

【美食谈】 民以食为天，美食是生活不可或缺的一部分，我们身体的健康与我...

妈，我刚在开车。没听见你打电话!

刚才我看了一篇文章，里面说现在的牛奶啊，里面抗生素超标严重，还有致癌物质，你跟娃赶紧都不要喝了。
我一会把文章给你发过去，你好好看看！赶紧停了……

妈，你刚发的文章我看到了，人家早都已经辟谣了！那个是谣言！

啥谣言！人家里面写的那么清楚，你喝不喝我们不管，赶紧给我孙子先把牛奶停了……
我不跟你说了，我赶紧给你姨妈和

Key Challenges in Software Engineering

1. Effective communication
   – Between the engineering team and other stakeholders
   – Within the engineering team

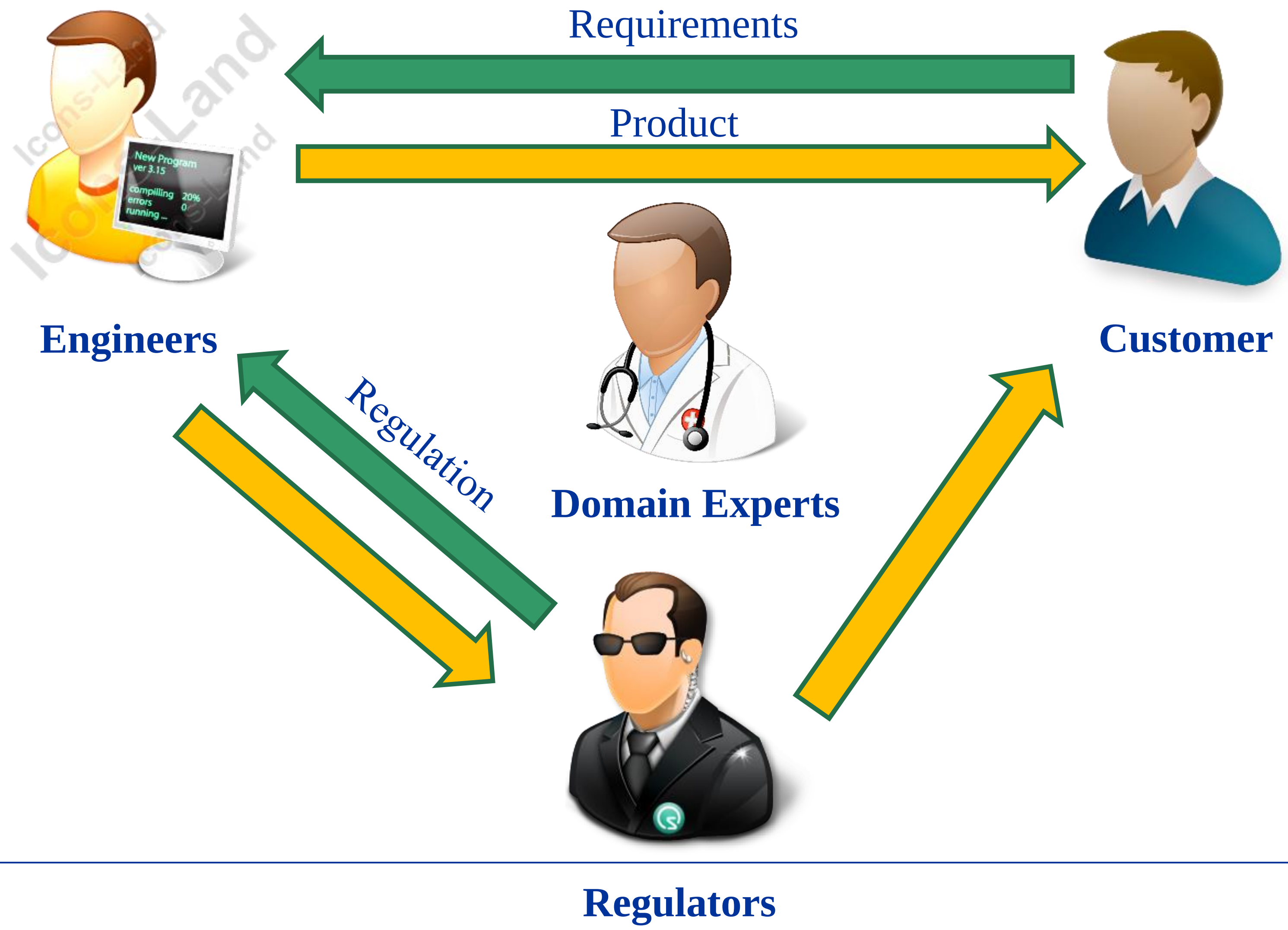
2. Risk Management
   – How to balance conflicting judging criteria?

3. Validation
   – How do you convince all stakeholders that the software is effective/safe/secure?

1. Effective communication

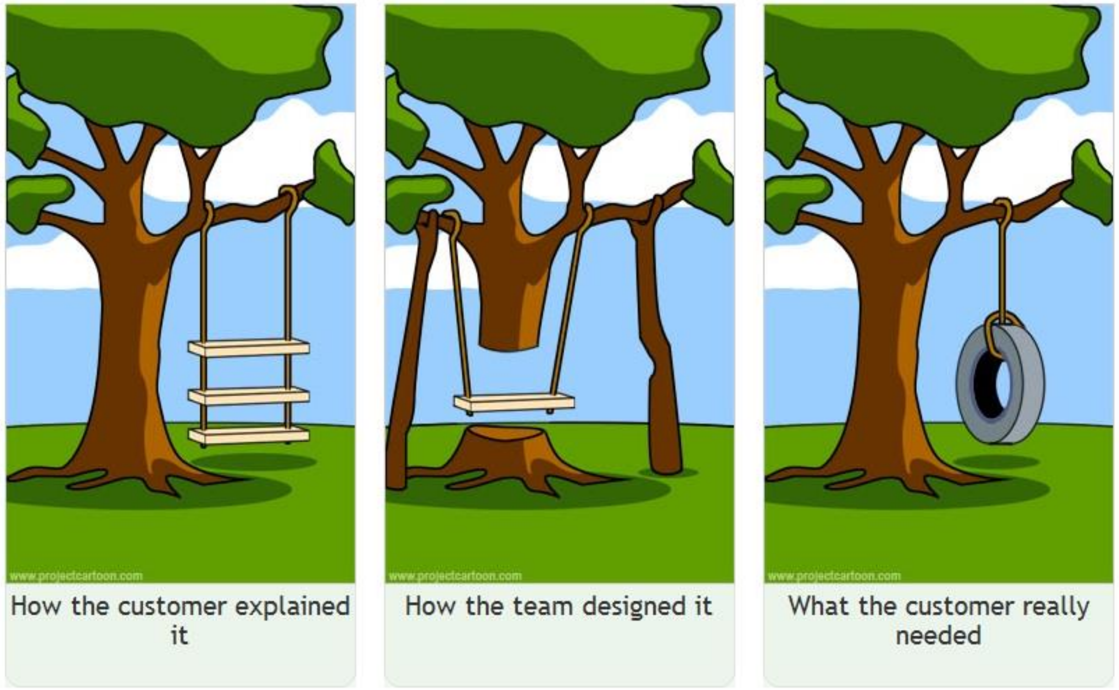
1. Effective Communication

Stakeholders for software



Miscommunications

- The customer fails to explain their needs well.
- The customer may not know what he/she wants
- The analyst may not understand the customer's need.
- Your analyst may not convey the requirements to the development team

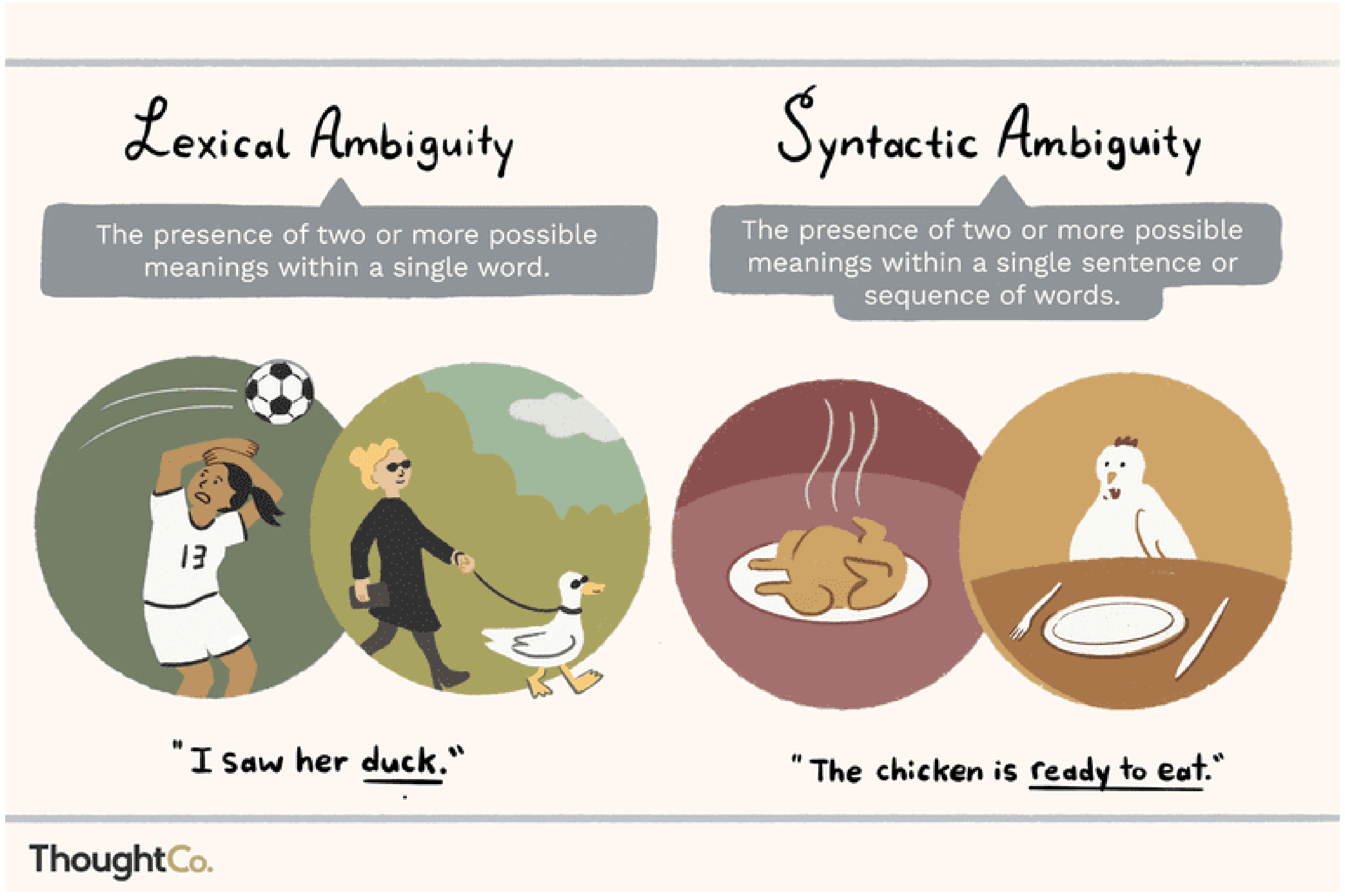


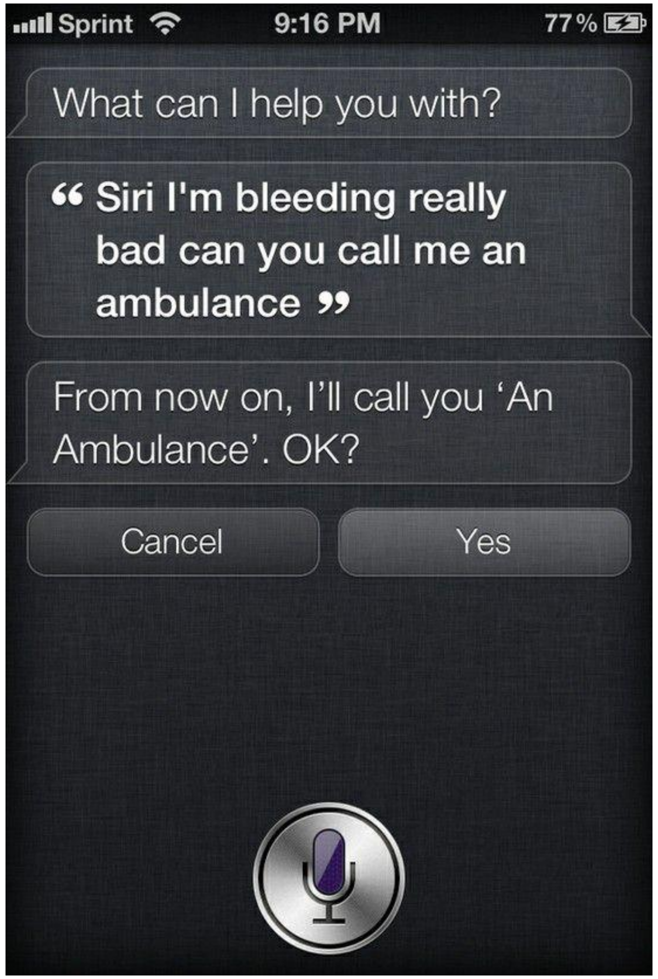
1. Effective Communication

Domain Knowledge: “The Expert”

Our company has a new strategic initiative to increase market penetration, maximise brand loyalty, and enhance intangible assets.

Natural Languages Are Prone to Ambiguities


Lexical Ambiguity
The presence of two or more possible meanings within a single word.
Syntactic Ambiguity
The presence of two or more possible meanings within a single sentence or sequence of words.
13
"I saw her duck."
"The chicken is ready to eat."
ThoughtCo.



What can I help you with?

“Siri I'm bleeding really bad can you call me an ambulance”

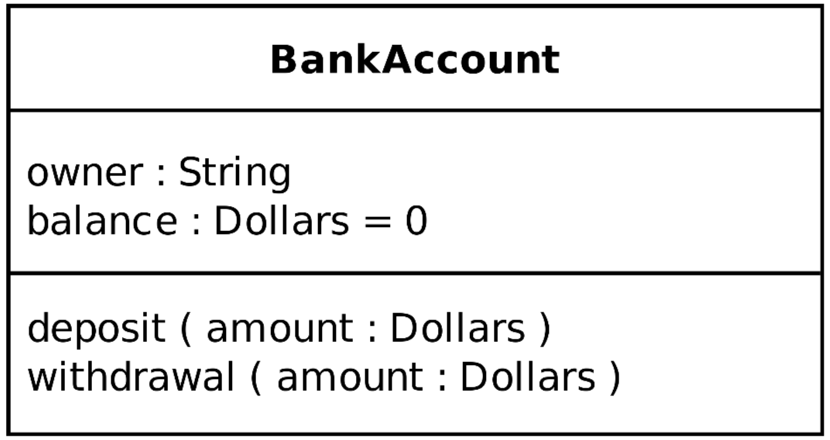
From now on, I'll call you 'An Ambulance'. OK?

Communications among various stakeholders

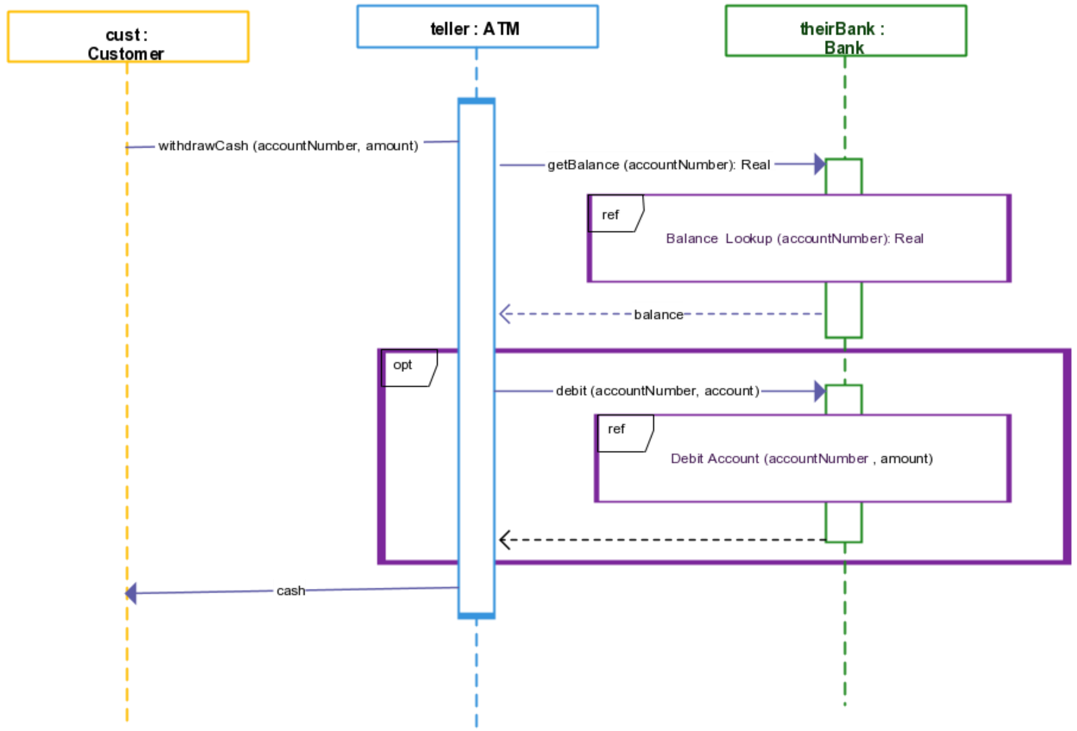
- Need a common language for communication
- Unified Modeling Language (UML)
- Recognized as an international standard
- It’s just a tool, not a solution

Examples of UML Diagrams

- Structural diagram

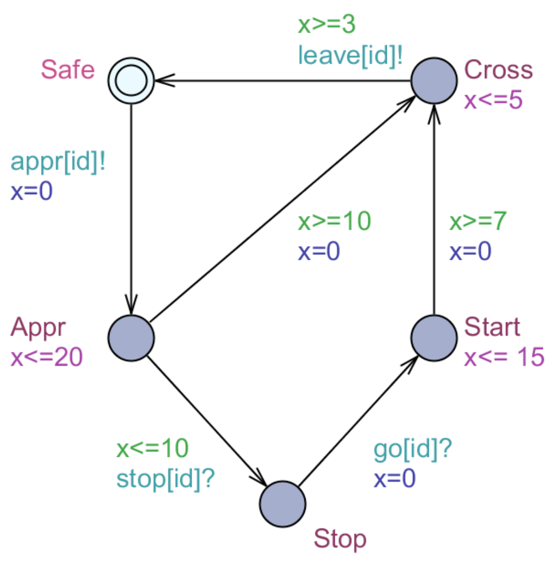


- Behavioral Diagram

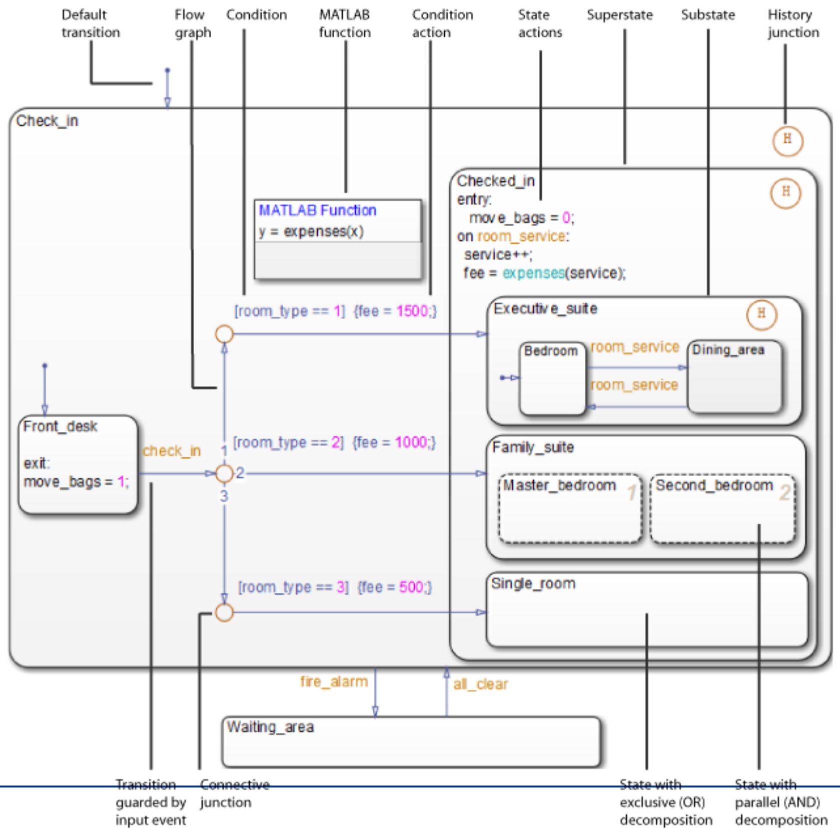

cust : Customer
teller : ATM
theirBank : Bank
withdrawCash (accountNumber, amount)
getBalance (accountNumber): Real
ref
Balance Lookup (accountNumber): Real
balance
opt
debit (accountNumber, account)
ref
Debit Account (accountNumber , amount)
cash

Communication within the team

- Formal models
  - Timed automata



- Simulation models
  - Simulink



The lifecycle of a successful software is long

- Other team members should understand your code and documentation

- Other team members should be able to easily “inherent” code from you

What we want you to learn in this course

- Communication skills are important no matter what you do in the future
  - Documentation
  - Meetings and presentations

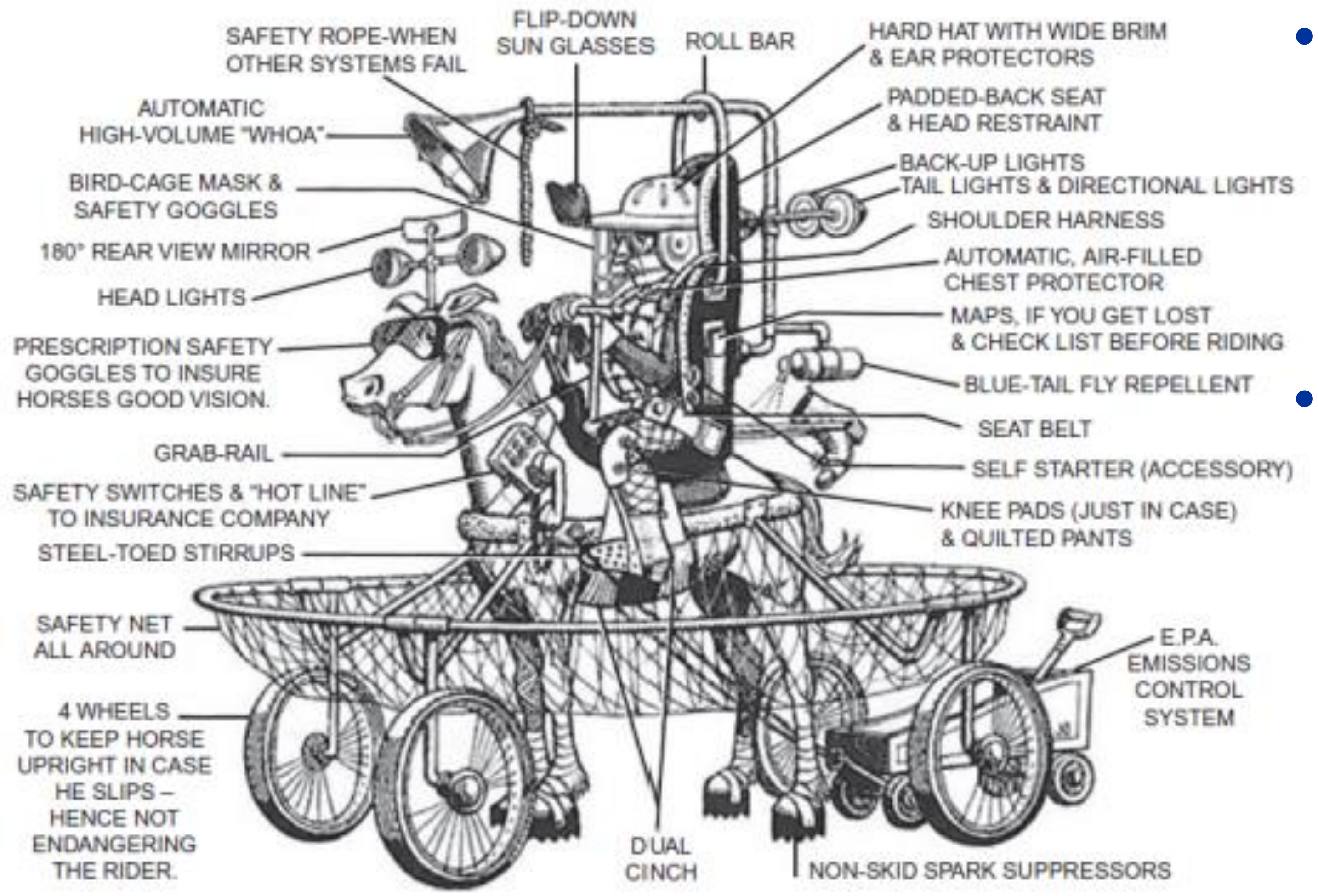
- Be mentally prepared
  - Respect other people’s domain expertise
  - Accept that other people may not know what you know

- How to analyze problems in other domains?

2. Risk Management

Are they good aircrafts?

The “All-around” Solution

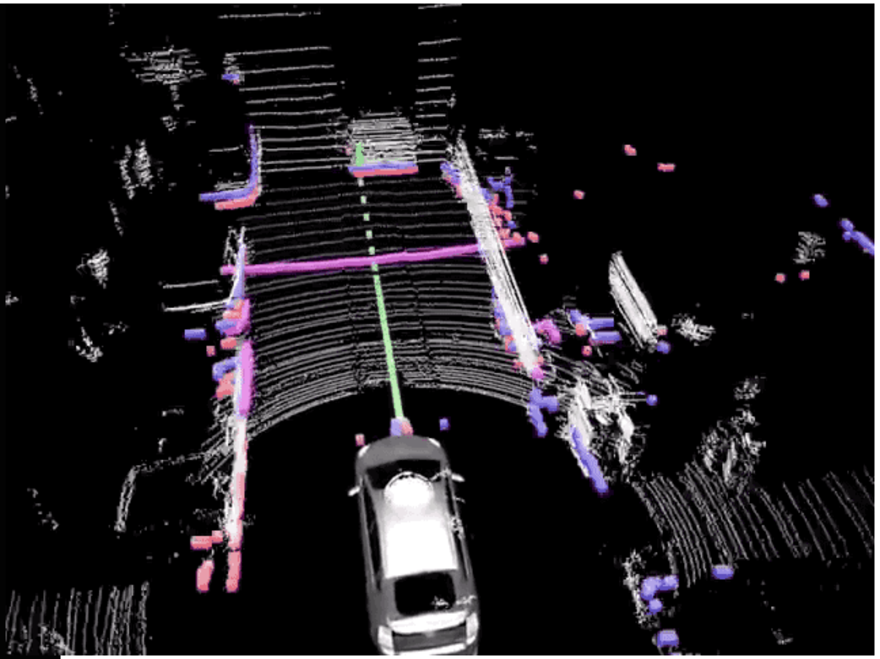

SAFETY ROPE-WHEN OTHER SYSTEMS FAIL
FLIP-DOWN SUN GLASSES
ROLL BAR
HARD HAT WITH WIDE BRIM & EAR PROTECTORS
AUTOMATIC HIGH-VOLUME "WHOA"
PADDED-BACK SEAT & HEAD RESTRAINT
BIRD-CAGE MASK & SAFETY GOGGLES
BACK-UP LIGHTS TAIL LIGHTS & DIRECTIONAL LIGHTS
SHOULDER HARNESS
180° REAR VIEW MIRROR
AUTOMATIC, AIR-FILLED CHEST PROTECTOR
HEAD LIGHTS
MAPS, IF YOU GET LOST & CHECK LIST BEFORE RIDING
PRESCRIPTION SAFETY GOGGLES TO INSURE HORSES GOOD VISION.
BLUE-TAIL FLY REPELLENT
SEAT BELT
GRAB-RAIL
SELF STARTER (ACCESSORY)
SAFETY SWITCHES & "HOT LINE" TO INSURANCE COMPANY
KNEE PADS (JUST IN CASE) & QUILTED PANTS
STEEL-TOED STIRRUPS
SAFETY NET ALL AROUND
E.P.A. EMISSIONS CONTROL SYSTEM
4 WHEELS TO KEEP HORSE UPRIGHT IN CASE HE SLIPS – HENCE NOT ENDANGERING THE RIDER.
DUAL CINCH
NON-SKID SPARK SUPPRESSORS

- Risk control measures may affect other system properties

- Justify that benefits outweigh the risks

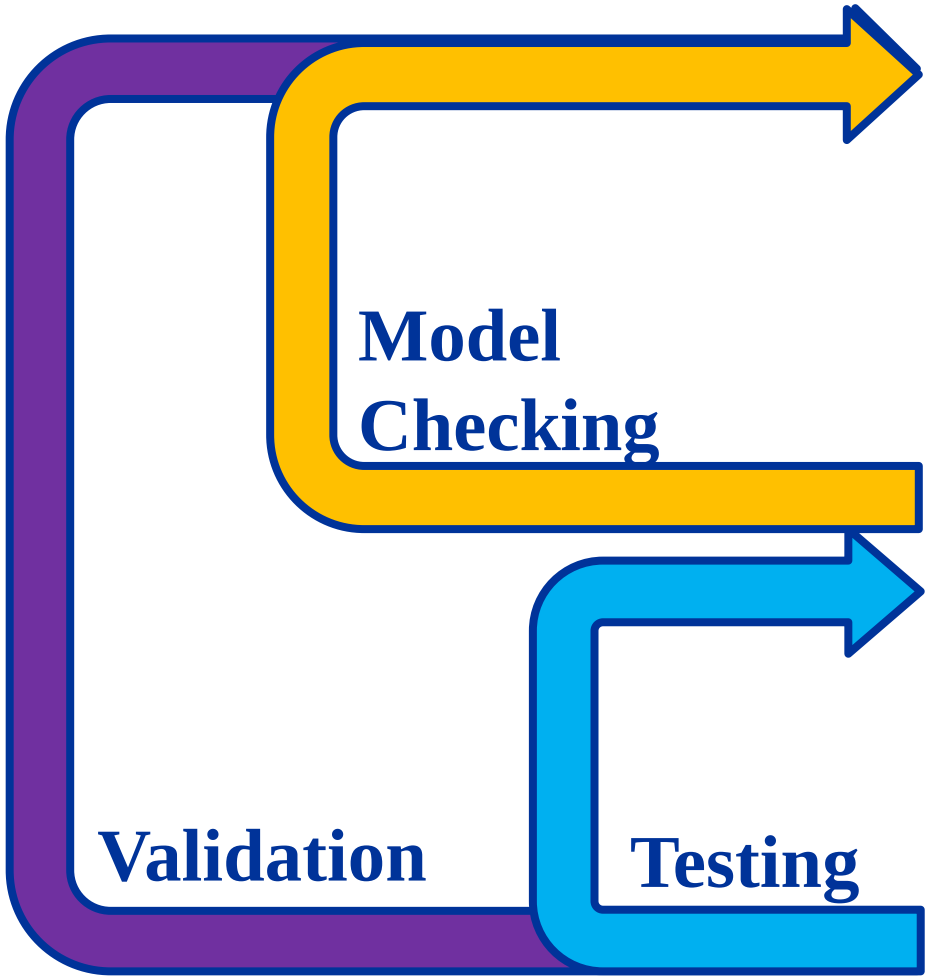
Balancing among risks

- Uber Autonomous Vehicle Accident (March 18, 2018)
- Pedestrian Identification: balancing false-positives vs. false-negatives



3. Validation

Software Development Process



- Requirements
  - i.e. the vehicle should not crash with any other objects
- Specifications
  - i.e. when the vehicle detects an object on its path, apply brake
- Implementation

What should we validate?

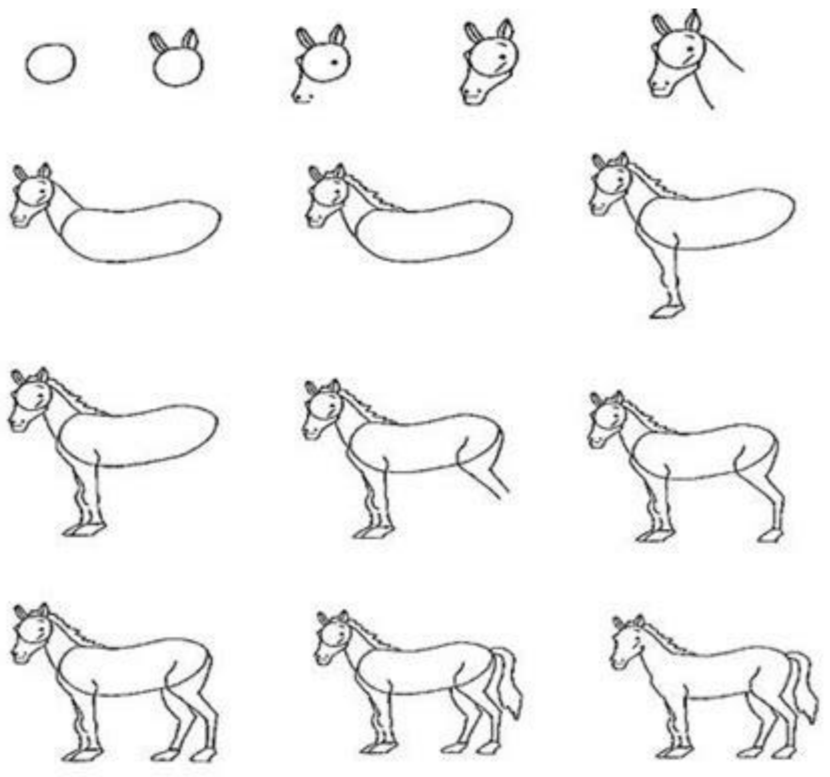
- Efficacy: The system can do its job as designed
- Safety: Under intended use, the system will not harm the user and its surroundings
- Security: Prevent malicious use of the system

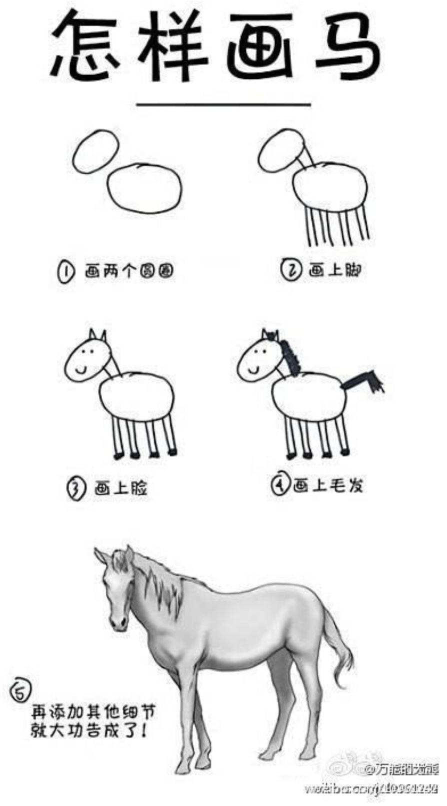
How to convince others that your system is good?

- Rigorousness of the development process
- Rigorousness of the techniques
- Demonstration of effort
- “All or nothing”

Iterative Software Development

- Develop "validatable" artifacts early





Model-based Software Design

- From verified model to verified code
  - Business model (in UML)
  - Analysis model (in UPPAAL)
  - Design model (in UML/Simulink)
  - Code (in Matlab)
- Verify analysis/design model
- Maintain traceability during development

Why Early Prototyping?

An analogy from the movie industry

Stakeholders

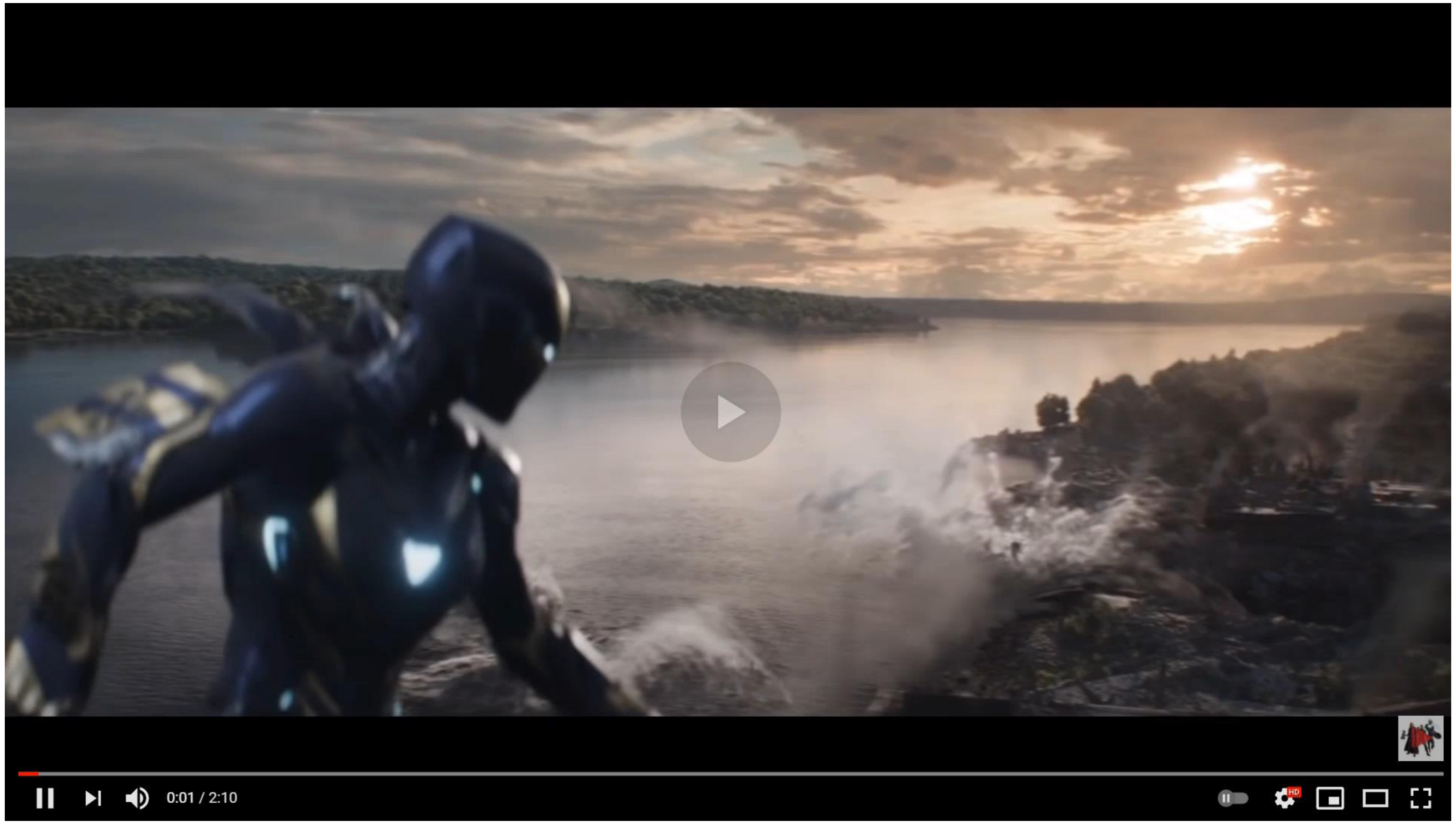
- Investor
- Production Team
  - Director
  - Actors
- Audience

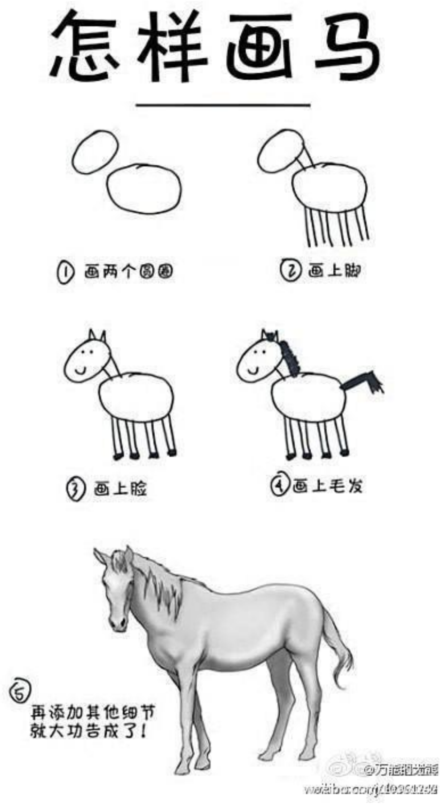
How can the production team convince the investor that they can make a good movie that makes profit?

Scripts – The Avengers Endgame

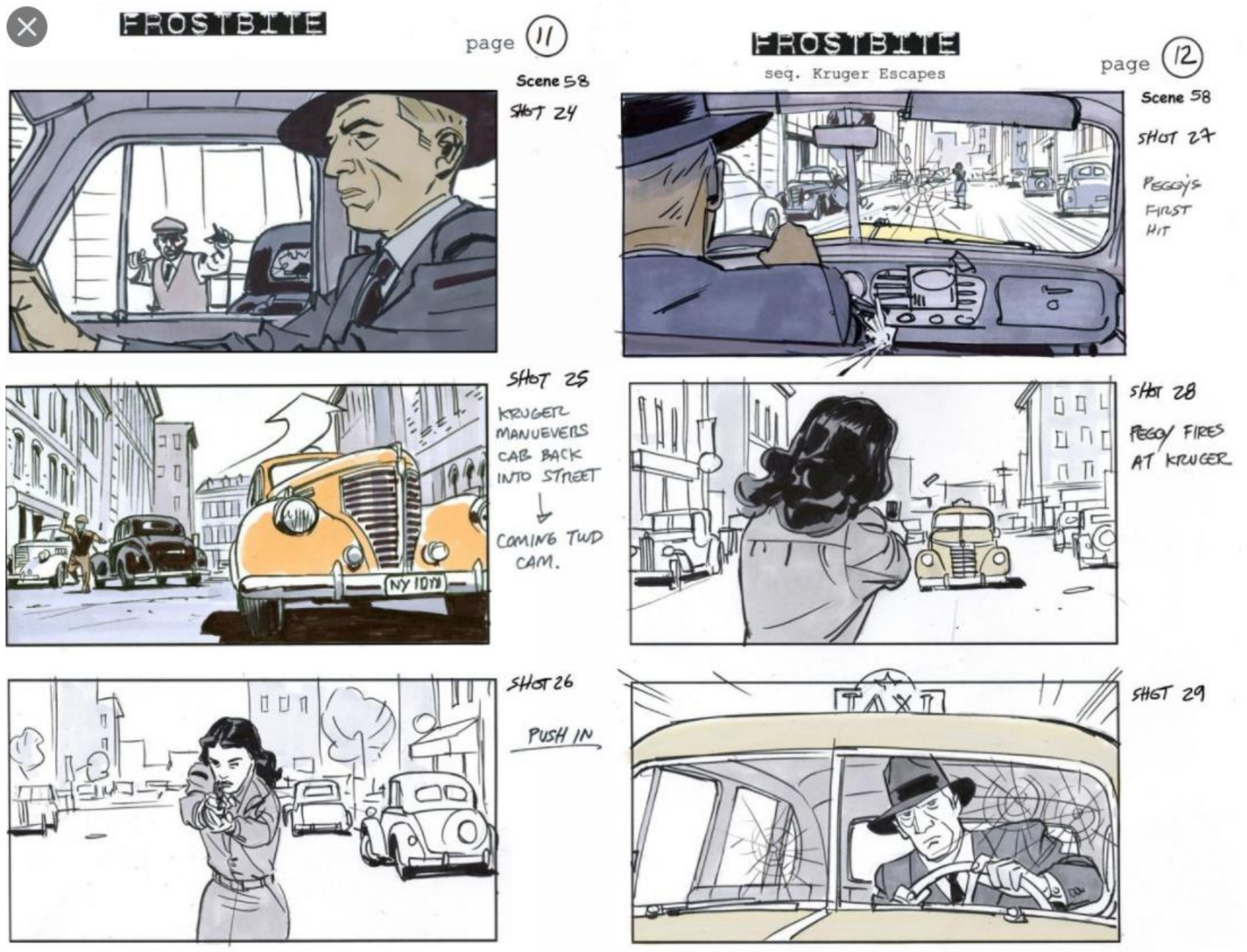
- The barrage destroys many on the battlefield
- The Barrage opens the riverbank and threatens to flood the battlefield, Dr. Strange and the other sorcerers have to hold the floodwater back
- Peter Parker and the Gauntlet is about to be overwhelmed by enemy forces BUT Steve hurls Mjolinir
- Peter catches a ride on it then with Valkyrie, BUT the ship's cannon fire knocks both Peter and Valkyrie to the ground
- The ship's fire is going to KILL THEM ALL. THE FIRE IS CLOSING IN ON THEM WITH NO ESCAPE. ALL IS LOST…

The Final Movie Clip



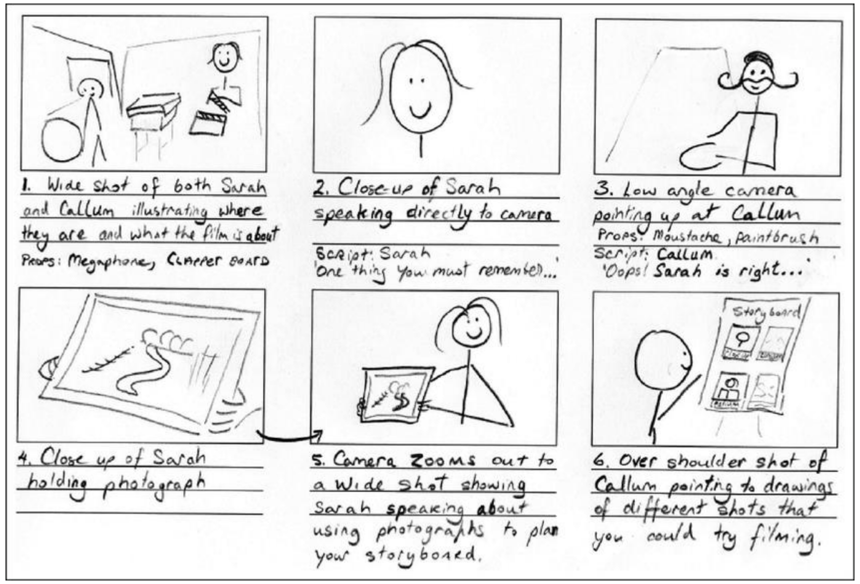


Storyboard


FROSTBITE
page 11
Scene 58
SHOT 24
SHOT 25
KRUGER MANUEVERS CAB BACK INTO STREET
COMING TWD CAM.
SHOT 26
PUSH IN
FROSTBITE
seq. Kruger Escapes
page 12
Scene 58
SHOT 27
PEGGY'S FIRST HIT
SHOT 28
PEGGY FIRES AT KRUGER
SHOT 29
TAXI
NY 1019

LUCKY STAR CAB COMPANY
0:00 / 3:58

More Storyboards



1. Wide shot of both Sarah and Callum illustrating where they are and what the film is about
Props: Megaphone, Clapper board

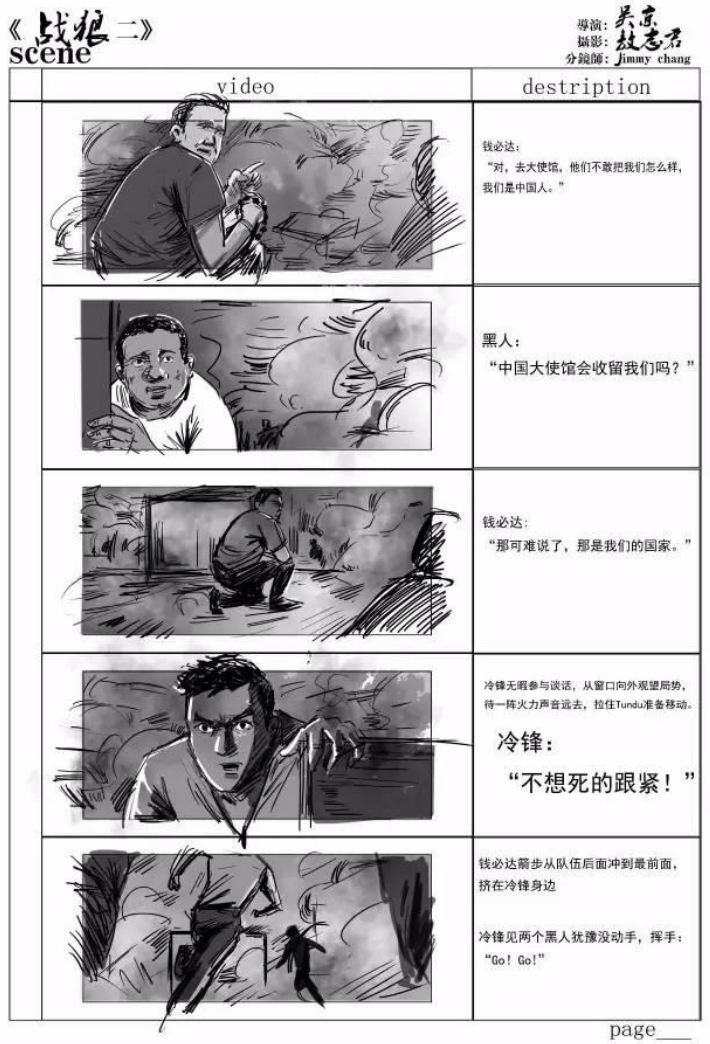
2. Close-up of Sarah speaking directly to camera
Script: Sarah
'One thing you must remember...'

3. Low angle camera pointing up at Callum
Props: Moustache, paintbrush
Script: Callum
'Oops! Sarah is right...'

4. Close up of Sarah holding photograph

5. Camera zooms out to a wide shot showing Sarah speaking about using photographs to plan your storyboard.

6. Over shoulder shot of Callum pointing to drawings of different shots that you could try filming.



《战狼二》
scene

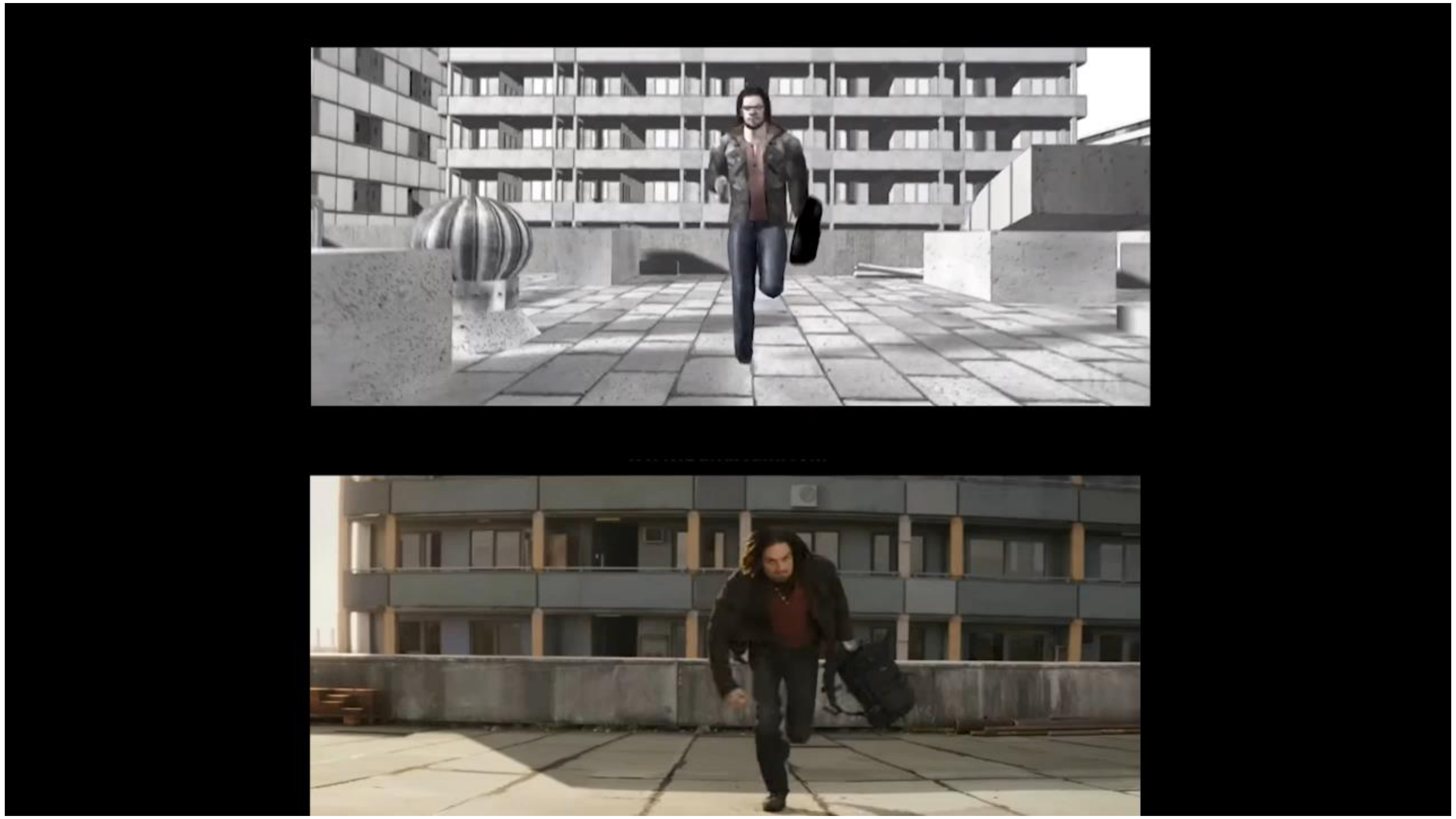
導演：吴京
攝影：赵志君
分鏡師：Jimmy chang

	video	destription
		钱必达： "对，去大使馆，他们不敢把我们怎么样，我们是中国人。"
		黑人： "中国大使馆会收留我们吗？"
		钱必达： "那可难说了，那是我们的国家。"
		冷锋无暇参与谈话，从窗口向外观望局势，待一阵火力声音远去，拉住Tundu准备移动。 冷锋： "不想死的跟紧！"
		钱必达箭步从队伍后面冲到最前面，挤在冷锋身边 冷锋见两个黑人犹豫没动手，挥手： "Go! Go!"

page___

Previs

Previs (Pre-visualization)



What we learned

- Quick prototyping
  - Gets feedbacks early
  - Saves money
  - Earns trust
- Just having a good idea is not enough
- Mastering the new tools is very important

The Analogy

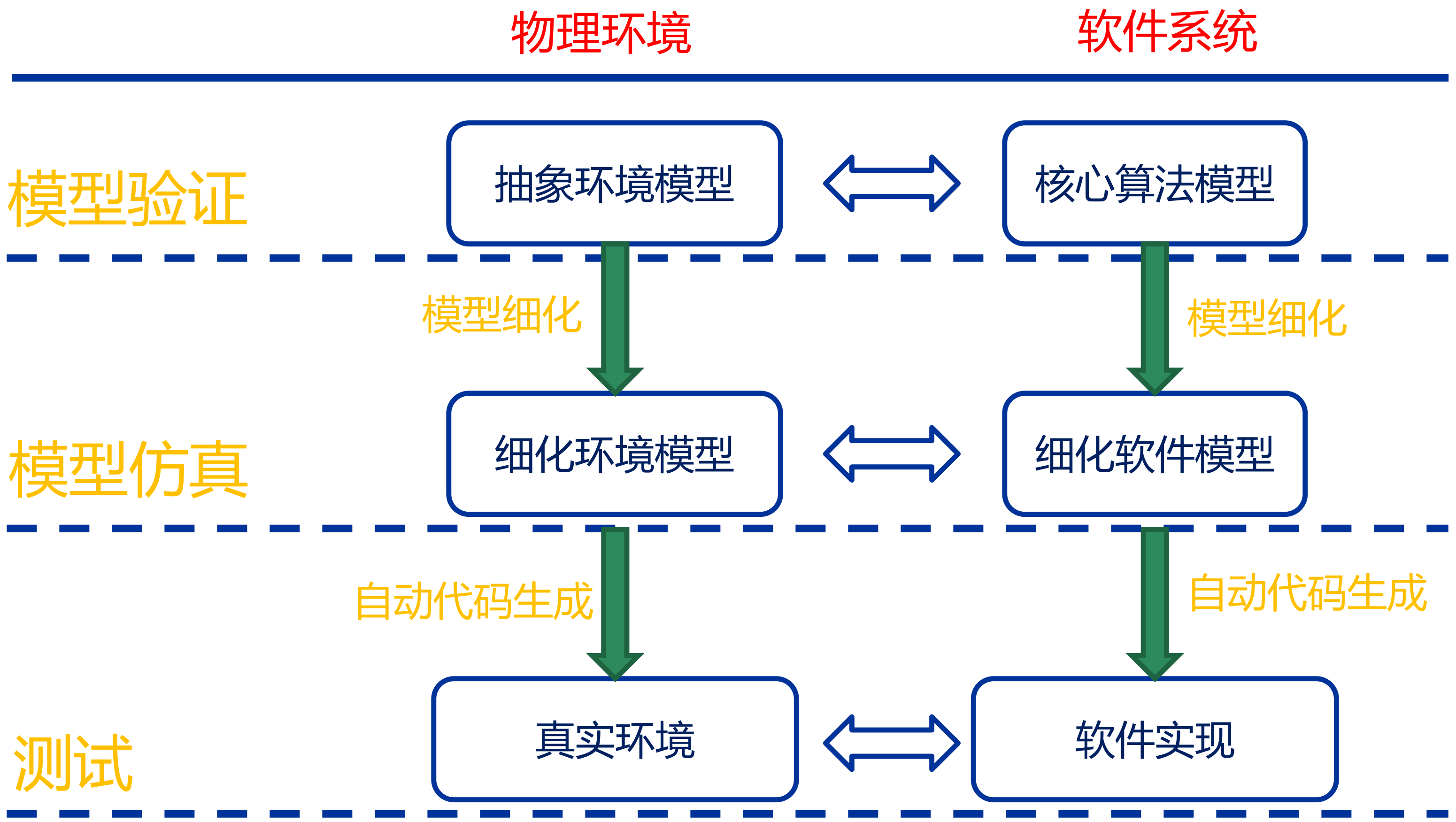
Movie Making

- Script
- Storyboard
- Previs
- Techvis

Software Development

- Requirement document
- UML
- Models->Prototypes
- Model translation & Code generation

Model-based Software Design



Key Challenges in Software Engineering

1. Effective communication
   – Between the customers and the engineering team
   – Within the engineering team

2. Risk Management
   – How to balance conflicting judging criteria?

3. Validation
   – How do you convince all stakeholders that the software is effective/safe/secure?

Curriculum

- Software development lifecycle
- Capture software requirements using UML
- Strike a balance: risk management
- Early bug-finding using model checking
- Maintain traceability in model-based software design
- Software testing

Grading

- Homework: 4*5%
- Midterm: 20%
- Final Project: 60%

Project Logistics

- 3 students per team (1,2,3)
- 3 mini projects (a,b,c)
- Each project has 3 stages
  - Requirement (R)
  - Development (D)
  - Validation (V)
- Student 1: a.R+b.D+c.V
- Student 2: b.R+c.D+a.V
- Student 3: c.R+a.D+b.V

Checkpoints and Progression

- Team meeting every week
  - Report on what has been done and plan for next week
  - Part of the demonstration of "effort"

- 3 Customer Consultations
  - Chance to demonstrate initial results and ask for feedback
  - Please take them seriously!

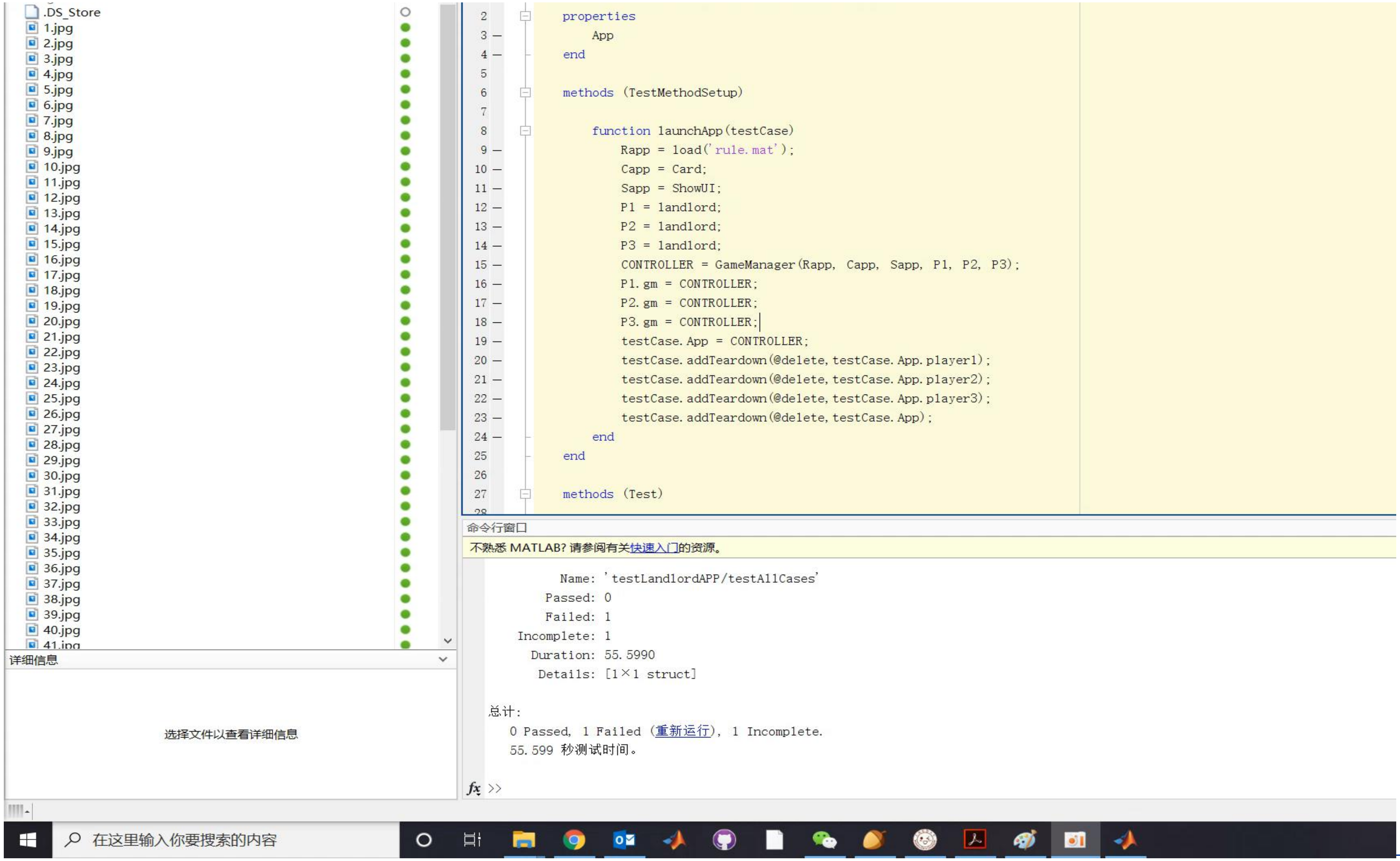
Project Grading

- Overall Product (70%+10%)
  - Functional requirements (40%)
  - Non-functional requirements (15%)
  - Validation (15%)
  - Extra Credits (10%)
- Documentation (30%)
  - Requirement (10%)
  - Development (10%)
  - Validation (10%)

Grading Example

Project 1			Project 2			Project 3		
Overall Product		55	Overall Product		60	Overall Product		65
Documentation	Requirement	8	Documentation	Requirement	5	Documentation	Requirement	6
	Development	7		Development	4		Development	3
	Validation	5		Validation	9		Validation	7

	Score
Student 1 (1R2D3V)	(55+60+65)/3+8+4+7=79%
Student 2 (2R3D1V)	(55+60+65)/3+5+3+5=73%
Student 3 (3R1D2V)	(55+60+65)/3+6+7+9=82%



```
.DS_Store
1.jpg
2.jpg
3.jpg
4.jpg
5.jpg
6.jpg
7.jpg
8.jpg
9.jpg
10.jpg
11.jpg
12.jpg
13.jpg
14.jpg
15.jpg
16.jpg
17.jpg
18.jpg
19.jpg
20.jpg
21.jpg
22.jpg
23.jpg
24.jpg
25.jpg
26.jpg
27.jpg
28.jpg
29.jpg
30.jpg
31.jpg
32.jpg
33.jpg
34.jpg
35.jpg
36.jpg
37.jpg
38.jpg
39.jpg
40.jpg
41.jpg
```

```matlab
    properties
        App
    end

    methods (TestMethodSetup)

        function launchApp(testCase)
            Rapp = load('rule.mat');
            Capp = Card;
            Sapp = ShowUI;
            P1 = landlord;
            P2 = landlord;
            P3 = landlord;
            CONTROLLER = GameManager(Rapp, Capp, Sapp, P1, P2, P3);
            P1.gm = CONTROLLER;
            P2.gm = CONTROLLER;
            P3.gm = CONTROLLER;
            testCase.App = CONTROLLER;
            testCase.addTeardown(@delete,testCase.App.player1);
            testCase.addTeardown(@delete,testCase.App.player2);
            testCase.addTeardown(@delete,testCase.App.player3);
            testCase.addTeardown(@delete,testCase.App);
        end
    end

    methods (Test)
```

命令行窗口

不熟悉 MATLAB? 请参阅有关快速入门的资源。

```
          Name: 'testLandlordAPP/testAllCases'
        Passed: 0
        Failed: 1
    Incomplete: 1
      Duration: 55.5990
       Details: [1×1 struct]

总计:
   0 Passed, 1 Failed (重新运行), 1 Incomplete.
   55.599 秒测试时间。

fx >>
```

详细信息

选择文件以查看详细信息

在这里输入你要搜索的内容

Working as a team

- Team up with someone you trust
- You are responsible for some task doesn’t mean you “only” need to complete the task
- “Make other’s job easier”
- “The Black Sheep” will get severe penalties

You may complain about…

- “The requirements are too vague!”
  - It’s not a bug, it’s a feature!
- “I did my part, why do I get penalized for what others didn’t do?”
  - Because you are on the same team, that’s why.
- Please do not ask
  - “Can we get/lose points if we implement/not implement a feature?”

Logistics

- Slides are released 1 day before each lecture on blackboard
- Important announcements are sent via emails
- Forum available on Piazza (piazza.com/shanghaitech.edu.cn/spring2022/cs132)
  - Ask questions regarding implementations, not ideas

Academic Ethics

- Homework should be done alone
- Do not share code/documents among teams
- Feel free to reuse code segments within the team
- Do not use code/documents from other sources (previous years' or online)
- Violators will receive severe penalties