不仅仅是测试

Jacky Guo

About this presentation

- Overview
  - Testing vs. Quality Engineering
  - Act as a PM or Dev

Objectives

- After this presentation, you'll be able to:
  - Identify a phased approach for quality engineering
  - Identify two actions for tomorrow to impact the quality

Test vs. Quality Engineering (SQE)

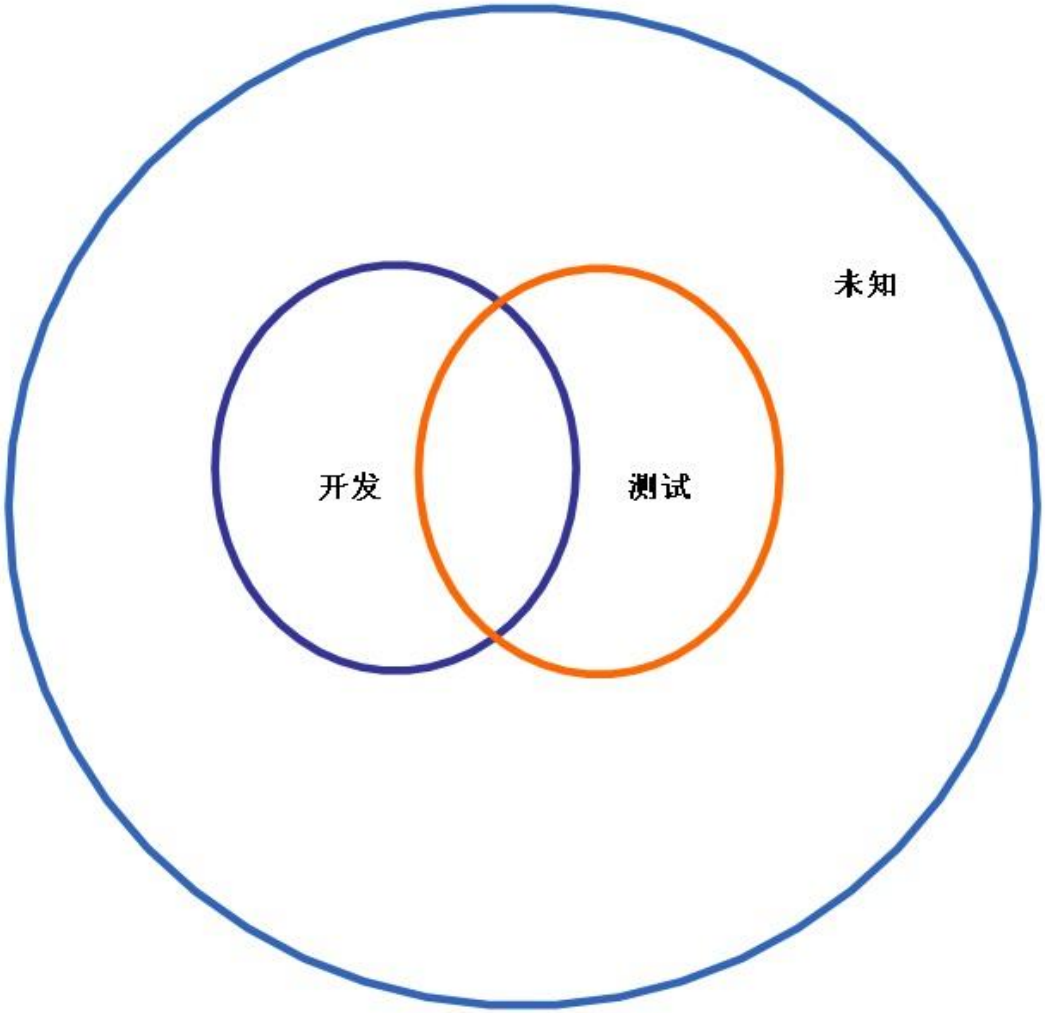
Test

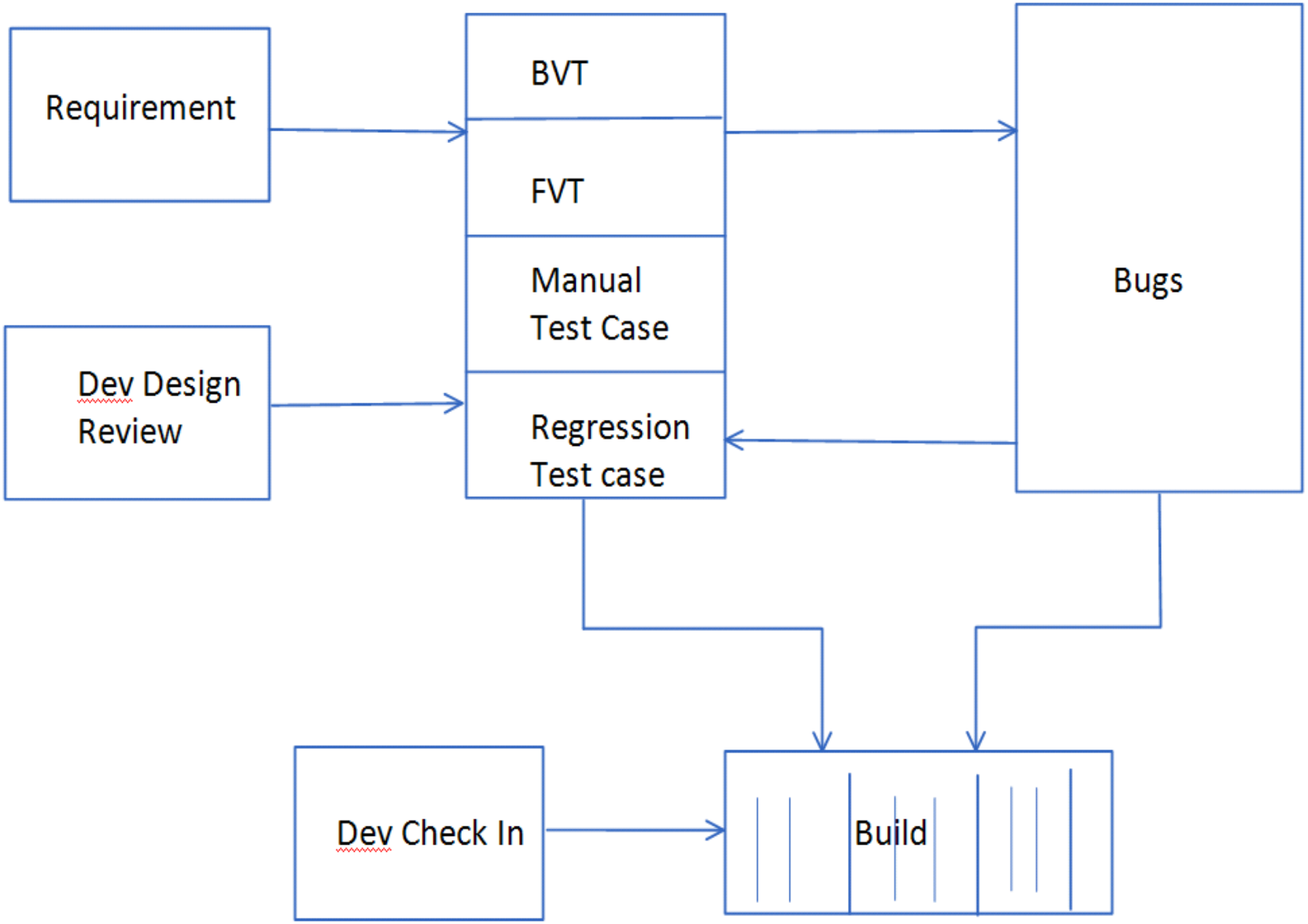
- Late in the timeline
- Reactive
- Find defects
- Singular in affect

Quality Engineering

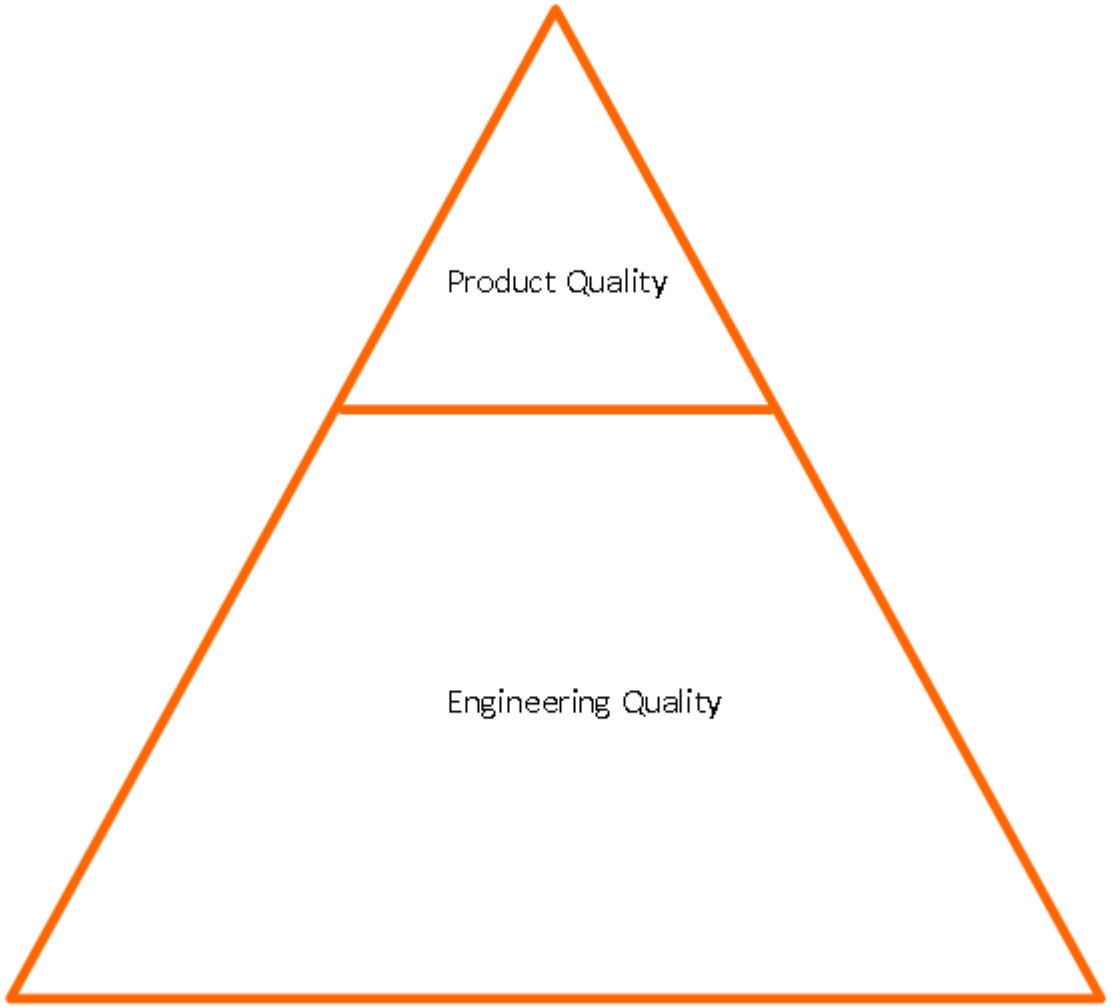
- First in the timeline
- Proactive and reactive
- Understand defects
- Multiplicative in affect

KNOWLEDGE



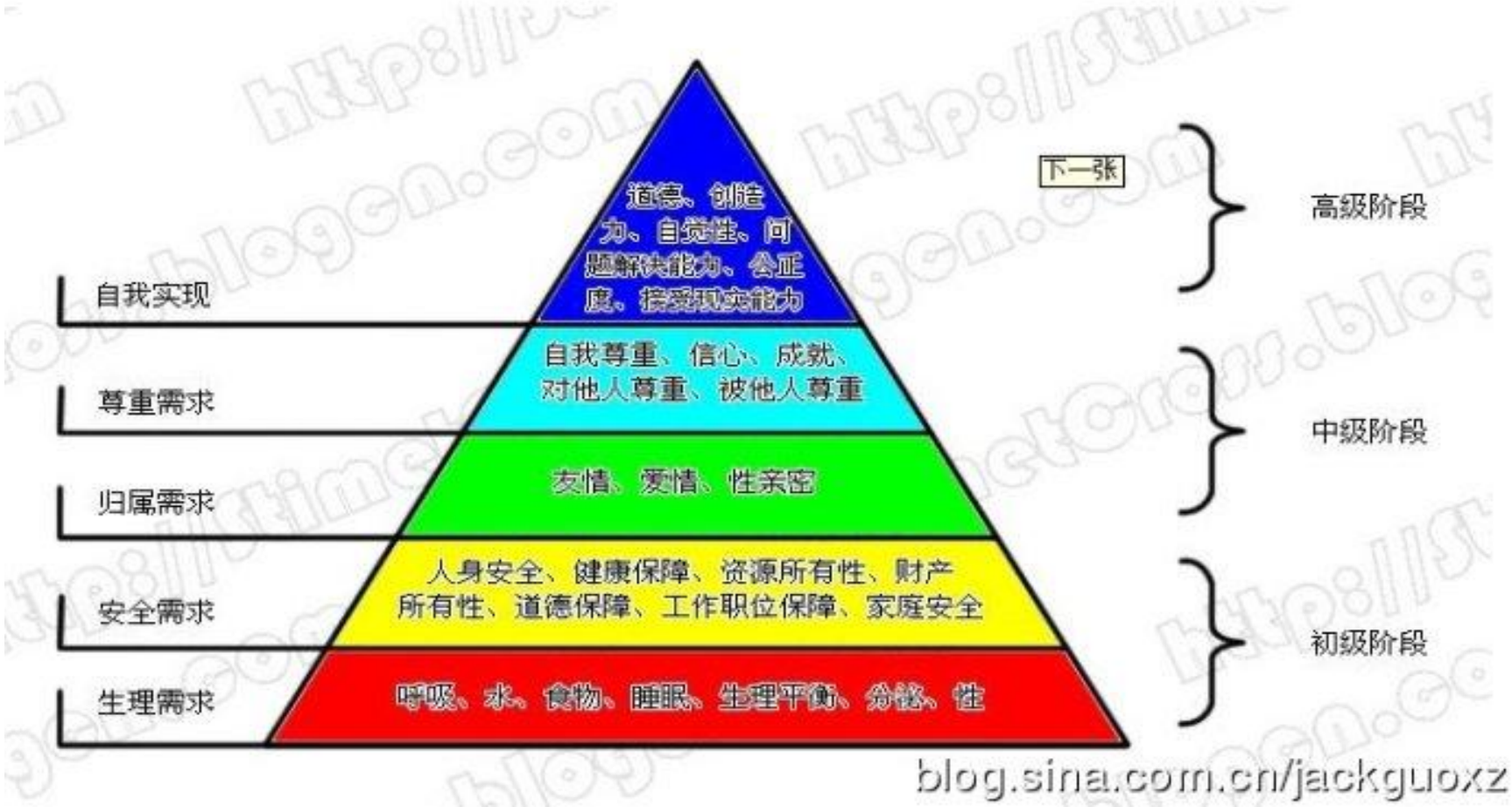

Requirement
Dev Design Review
BVT
FVT
Manual Test Case
Regression Test case
Bugs
Dev Check In
Build

What is Quality



Phase 1: how do I start?

- Written requirements
  - Clarify user& user scenario
  - Have your design principle
- Simple
  - One Step instead of two or three
  - Avoid repeated things
  - Batch command
- Bug the documents
  - "Does everyone see the same picture?"
  - "Will" versus "might, should, desirable"
- Testability/ Sustainability
- SMART


下一张
道德、创造力、自觉性、问题解决能力、公正度、接受现实能力
自我实现
自我尊重、信心、成就、对他人尊重、被他人尊重
尊重需求
友情、爱情、性亲密
归属需求
人身安全、健康保障、资源所有性、财产所有性、道德保障、工作职位保障、家庭安全
安全需求
呼吸、水、食物、睡眠、生理平衡、分泌、性
生理需求
高级阶段
中级阶段
初级阶段
blog.sina.com.cn/jackguoxz

Case Studies

- Weibo
- 360
- Douban
- Alipay
- Dianping
- Online Game


西
447
W
新闸路
Xinzha Rd.
东
403
E
HC360慧聪网


淘宝会员
支付宝会员
为了您的账号安全，请输入验证码。
账户名
@hotmail.com
密 码
***
验证码
M5N
看不清
换一张
安全控件登录
登 录
忘记密码?
使用手机号码登录
免费注册
已经购买过的访客，点此登录


48
49
43
45
40
41
37
32
33
35
28
29
30
23
26
27
20
21
22
17
18
19
1
15

Phase 1: how do I start? (continued)

"Look" at Dev Design

- Understand dev design
  - ✓Work flow
  - ✓Data flow
  - ✓Data Structure
- Cover different user scenario
- Testability
- Risk

Phase 1: how do I start? (continued)

"Look" at code

- API & Parameters
- Code inspections or reviews
  - Reviews are light-weight
  - Target code subsets
  - Inspect only the most critical/difficult sections
- Debugging skill

我的Windows 7多次蓝屏，每次出现蓝屏的操作都不一样，比如这次是打开一个程序，下次是打开另外一个程序；最近实在受不了了，只好下载了windbg来分析一下dump文件（在C:\Windows\Minidump下面）；

- 到这里下载windbghttp://msdn.microsoft.com/zh-cn/windows/hardware/gg463009/ 或者这里 http://msdn.microsoft.com/zh-cn/windows/hardware/gg463016
- 打开dump文件，然后运行 !analyze -v

```
ADDITIONAL_DEBUG_TEXT:
Use '!findthebuild' command to search for the target build information.
If the build information is available, run '!findthebuild -s ; .reload' to set symbol path and load symbols.

MODULE_NAME: Hooksys
FAULTING_MODULE: 8403e000 nt
DEBUG_FLR_IMAGE_TIMESTAMP: 4edda48b
BUGCHECK_STR: 0x7f_8
CUSTOMER_CRASH_COUNT: 1
DEFAULT_BUCKET_ID: VISTA_DRIVER_FAULT
CURRENT_IRQL: 0

LAST_CONTROL_TRANSFER: from 840ce18e to 840cd1f0
```

- 然后运行lmvm Hooksys

```
start    end        module name
9b428000 9b450a80   Hooksys   T (no symbols)
    Loaded symbol image file: Hooksys.sys
    Image path: \??\C:\Windows\system32\drivers\Hooksys.sys
    Image name: Hooksys.sys
    Timestamp:        Tue Dec 06 13:13:47 2011 (4EDDA48B)
    CheckSum:         0002ED85
    ImageSize:        00028A80
    Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4
```

然后google了一下，发现hooksys.sys是瑞星实时监控所要使用的文件,与运行的文件密切相关，解决办法就是把瑞星杀毒软件卸载或者删除掉。

Phase 1: done

Congratulations

- You have just grabbed low-hanging fruit
- You have prevented countless bugs
- You have risen above at least 50% of the Industry

Now what?

Phase 2: change the process

Learn about your bugs

- Find the cause
  - Unexpected customer requirement?
  - Vague spec?
  - Rushed design?
  - Domain knowledge?
  - Code error?
    - Be gentle, accurate and honest
- Make a change
- Bug “Social Network” & Data mining

Call to action

- Start small, get a win
  - What are people most upset about?
  - What is the low-hanging fruit?
  - What does the RCA data say?
- Be prepared to set the example

Call to action (continued)

What can you do tomorrow (pick 1)?

- Document one requirement
- Document one design
- Code-review one critical section
- RCA 10 bugs
- Implement 10 unit tests

Q&A

SUMMARY

Test case Management

- TC from Bugs by customer/external users
- TC from review dev design
- TC from user scenario
- TC from Bugs review in case of regression
- Regression Test strategy("diff" VS "full")
- BVT runs before Dev Check in

SUMMARY

Bug Management

- Map to TC
- Bug quality
- Root cause(process may needs change)
- Bug fix review to evaluate the risk
- Control the bar to fix bug(potential risk)

Goals

1. Does the proposed design meet the *spec requirements* and features?
2. How can the proposed design be done in the *available time* in the milestone?

Steps

1. **Validate Design**.
   Are the algorithms and data structures appropriate for the problem?
2. **Validate Tasks**.
   Are the tasks complete for each feature?
3. **Validate Schedule**.
4. **Validate Dependencies**. Are the dependencies complete and understood?
5. **Validate Check-ins**. Are the scheduled check-ins testable and frequent enough for test to be efficient?
6. **Mark Open Issues**. At each validation step, tasks that have open issues that keep the task from being "codable" will have the Open Issues checkbox marked.

Code Review Checklist

General

- [ ] Code has been run through static analysis tools to remove top tier defects prior to review.
- [ ] All resources are efficiently created and properly disposed of.
- [ ] Look for logic errors
- [ ] Look for design patterns \ refactoring for optimization
- [ ] Look for extensibility, scalability...
- [x] Code should be buddy built, and provide a sanity machine for test if possible

Security

- [ ] Code is free of buffer-overruns.
- [x] Code satisfies the requirements of the Secure Windows Initiative (SWI) team.
- [ ] All input parameters must be validated and checked for null, as well as in Constructor

Internationalization

- [x] All user-accessible strings are stored in a localizable format, such as resource files.
- [x] All controls that display strings on the UI are sized to accommodate the largest translations.
- [ ] All code satisfies team internationalization guidelines, including multi-language input and multi-cultural data formats.
- [ ] All strings would be read by end users should be localized

Coding Standards

- [x] The code adheres to team coding standards, and exceptions are documented and justified.
- [x] Catch and handle exceptions correctly
- [x] Complete and clear comments in code, especially the algorithm you are using
- [x] Avoid hard-coded elements in your code, and identify the constants correctly

Reliability

- [x] All arguments on publicly accessible APIs are validated (includes protected for extensible classes).