Test Driven Development

Test First

Test First

- **Unit test**:
  het testen van een method in een klasse die je bouwt, terwijl je die klasse aan het bouwen bent.

- **Refactoring:** verbeteren van het ontwerp van bestaande code zonder de functionaliteit te wijzigen (wikipedia)

Test First

- **Test Fixture** of **Test Class**: een klasse met unit tests.
- **Test** of **Test Method**: een test in een Test Fixture.
- **Test Suite**: een set van gegroepeerde tests.
- **Test Harness** of **Test Runner** of **Test Framework**: tool die de tests uitvoert.

Test First: red - green - refactor

1. Schrijf een test
2. Run de tests, de nieuwe test faalt
3. Schrijf code die de test doet slagen
4. Run de tests, de nieuwe slaagt
5. Refactor de code

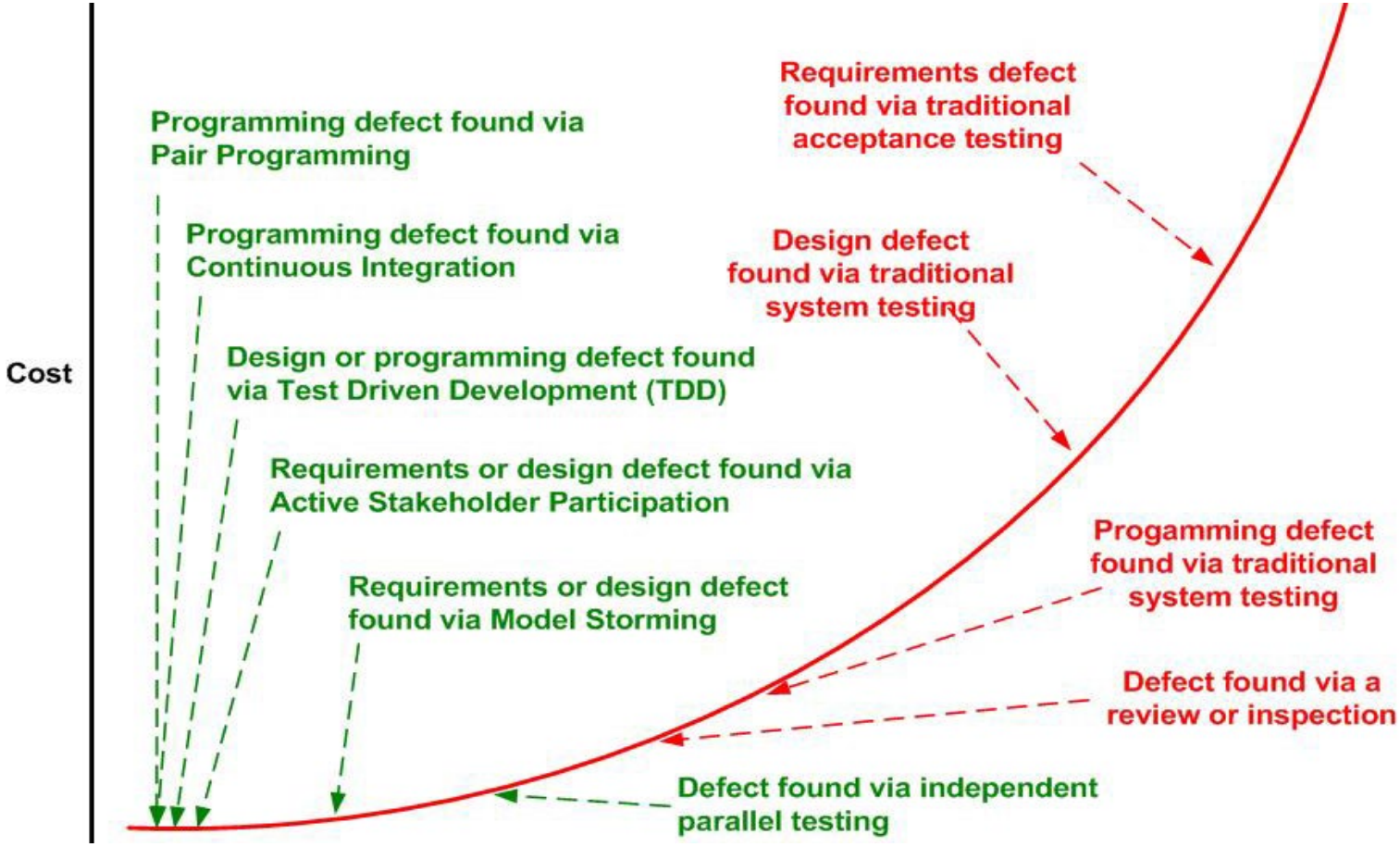
Test First: red - green - refactor

- Herhaal de cyclus: start met een nieuwe test om de functionaliteit van de klasse verder te brengen.
  De grootte van de stappen is variabel: indien de code om de test te doen slagen niet correct is, dan is een kleinere stap waarschijnlijk aangewezen.
- Neem zeker in het begin kleine stappen (wij beginnen met baby-stapjes).

Test First: waarom?

- Tests houden je uit de debugger
- Tests reduceren bugs in nieuwe features
- Tests reduceren bugs in bestaande Features
- Tests reduceren de "Cost of Change" (zie slide)
- Tests verbeteren het design van je klassen
- Tests zorgen voor mogelijkheden tot refactoring
- Tests beperken onnodige features
- Tests beschermen tegen "andere programmeurs"
- Testen **is leuk**
- Testen forceert je om halt te houden en na te denken
- Testen zorgt voor sneller ontwikkelen
- Testen **reduceert angst** voor wijzigingen en refactoring

Cost Of Change

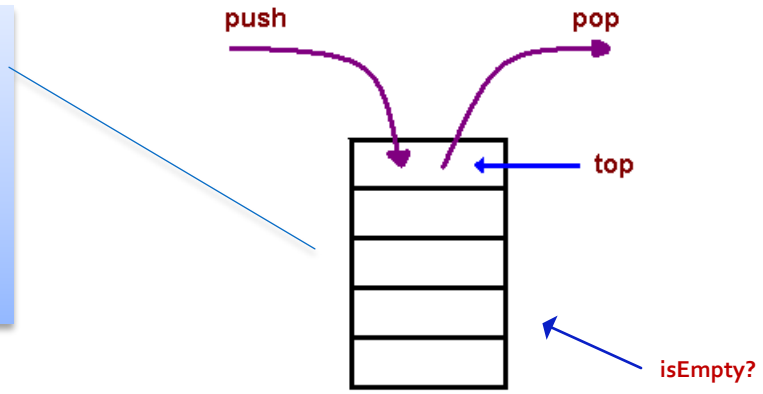


http://www.agilemodeling.com/essays/costOfChange.htm

Test First: Unbounded Stack

- We bouwen een klasse UnboundedStack.
- Een Stack (Stapel) is een datastructuur met de operaties push (duwt een element op de stapel) en pop (haalt het bovenste element van de stapel).

Stack met Push, Pop, Top en IsEmpty. LIFO principe (Last In First Out)



Test First: Test List na brainstorm

- Creëer een Stack en verifieer IsEmpty is True
- Push een element op de Stack en verifieer IsEmpty is False.
- Push een element, Pop het element, verifieer IsEmpty is True.
- Push een element, vraag Top op, verifieer IsEmpty is False
- Push een element, verifieer Top is dat element
- Pop van een lege Stack geeft fout
- Top opvragen van een lege Stack geeft fout
- Push meedere elementen, Pop, verifieer IsEmpty is False
- Push meedere elementen, verifieer Top is het laatste element
- Push meerdere elementen, Pop alle elementen, verifieer de volgorde met Top

Test First: eventuele bijkomende testen

Bij het testen en bouwen kunnen er nieuwe kandidaat testen opduiken.
Volgende testen zijn het gevolg van de discussie of we null willen toestaan als element (dat was een ja).

- Push null verifieer isEmpty is false
- Push null, verifieer Pop geeft geen fout
- Push null verifieer Top is null