

# SOFTWARE TESTEN SCHATTEN

*Schatting  
2 werkuur*

Hoofdtitel: Software Testen  
Subtitel: Schatting en Planning  
Auteur / docent: Vincent Van De Walle  
Versie: Gebruikt tijdens opleiding

© Syntra West

*De docent/auteur en Syntra West behouden zich het recht voor de inhoud, het beeldmateriaal en alle verdere informatie in deze PowerPoint zonder voorafgaand bericht aan te passen of te verbeteren.*

*Deze PowerPoint is met zorg samengesteld, waarbij alles werd in het werk gesteld om de juistheid van de inhoud te verzekeren, niettemin wordt deze PowerPoint ter beschikking gesteld zonder enige garantie.*

*De docent/auteur en Syntra West kunnen niet verantwoordelijk noch aansprakelijk gesteld worden voor eventuele materiële schade of andere vergissingen die voortvloeien uit het gebruik van de informatie uit deze PowerPoint of voor eventuele fouten of vergissingen in de PowerPoint. Dit cursusmateriaal is auteursrechtelijk beschermd.*

**Alles** uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, elektronisch of op welke andere wijze ook, of in een retrieval systeem worden opgeslagen, zonder voorafgaandelijke schriftelijke toestemming van de docent/auteur.

**Niets** uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, elektronisch of op welke andere wijze ook, of in een retrieval systeem worden opgeslagen, zonder voorafgaandelijke schriftelijke toestemming van de Syntra West.

leren. durven. doen.



# VINCENT VAN DE WALLE

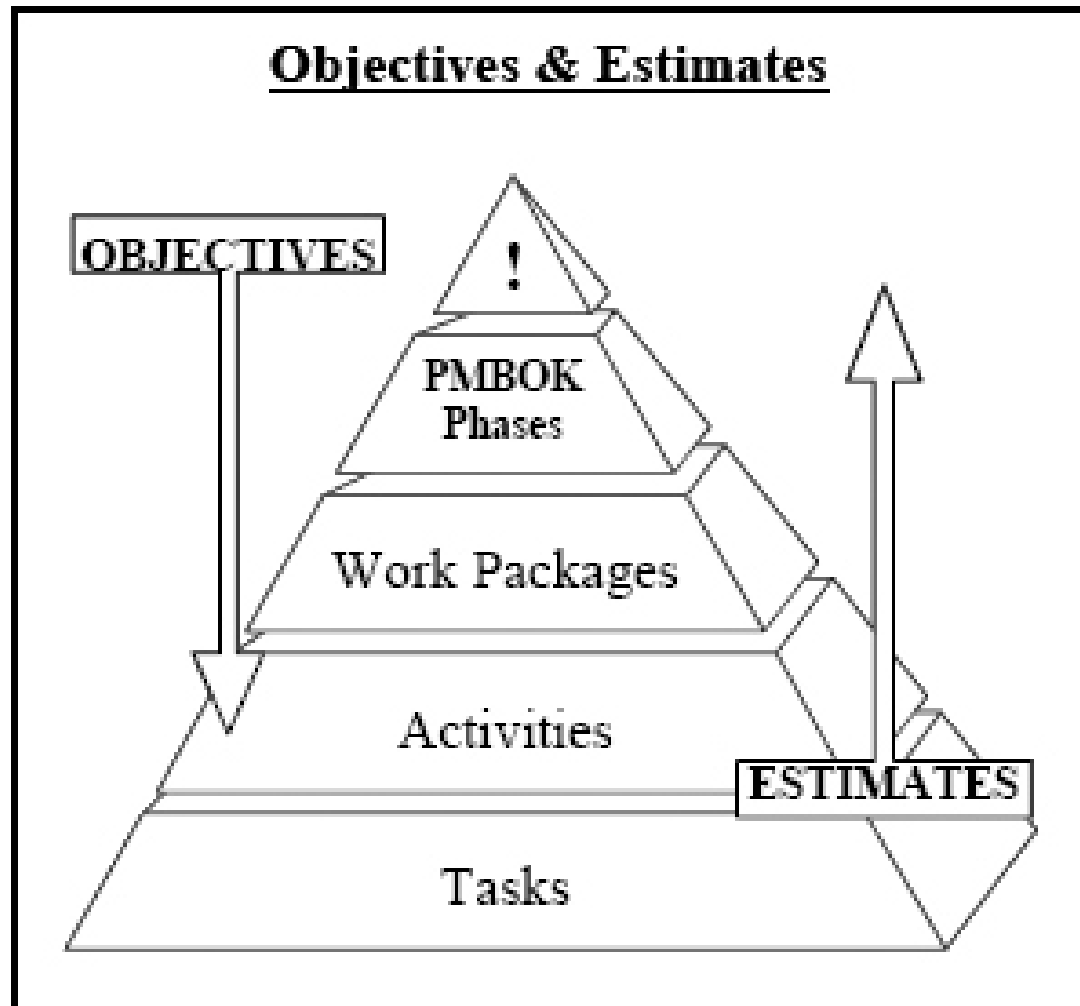
*Vincent@CopyPaste.be*  
*VincentVanDeWalle@digi4u.be*

+32 (0) 475 / 97.63.80

# Schattingen zijn onafhankelijk van objectieven

- **ALS HET DAT NIET IS, DAN VRAAG JE OM PROBLEMEN.**
  - Je mag nooit schattingen laten afhangen van de belangrijkheid van een item / taak / User Story.
- **ALS DE SCHATTINGEN TE HOOG ZIJN, OM DE OBJECTIEVEN UIT TE VOEREN, DAN MOET JE DE BELANGRIJKHEID LATEN SPELEN**
  - Niet omgekeerd
  - Schattingen nooit aanpassen aan de objectieven

# Schattingen zijn onafhankelijk van objectieven



# Waarom User Stories in Story Punten?

- **JE WEET NIET WIE HET GAAT DOEN.**
  - Het voorbeeld met Excel totalen.
  - Het voorbeeld van de landen.
  - Het schilderen van de vier kamers.
  - Senior – Junior in een bepaald kennis gebied.
- **0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, INFINITY, TAS KOFFIE**
  - Waarom deze cijfers?
  - Je vermijdt dat je exact zou willen schatten.
  - Het team creert voor zichzelf een marge.

# Waarom taken in uren?

- **ALS EEN TAAK GEASSIGNEERD WORDT, WEET JE WIE HET GAAT DOEN.**

- En dan kun je beter schatten
- Een taak heeft een kleinere grootte orde dan een User Story.
- Val niet in de valkuil van repetitief werk:
  - In software bestaat dat niet.
  - Als het wel bestaat, ben je geen goede software aan het schrijven.
  - Een herkenbaar voorbeeld.
    - Lezen, toevoegen, wijzigen, verwijderen van iets.

# Waarom werken we iets 90% af

- **NIEMAND HOUDT VAN DE PUNTJES OP DE “I” TE ZETTEN**
  - Dit is enorm tijdrovend.
  - Je krijgt de indruk dat dit geen “value” oplevert.
- **DIT IS 1 VAN DE “WASTES” UIT LEAN**
  - Onafgewerkte items.
  - Hoe kleiner de taak, hoe kleiner de kans dat iets niet wordt afgewerkt.
  - Hoe kleiner de User Story, hoe kleiner de kans dat iets niet wordt afgewerkt.



# Waarom schatten we fout?

- **WE VERGELIJKEN MET ANDERE TAKEN.**
  - Dit is enorm tijdrovend.
  - Je krijgt de indruk dat dit geen “value” oplevert.
- **WE GAAN ERVAN UIT DAT ER GEEN PROBLEMEN GAAN OPDUIKEN.**
  - Iedereen liegt (of vertelt halve waarheden)
  - Iedereen is altijd veel te optimistisch
  - De MVP ontwijkt alle problemen
    - En dan beginnen de problemen
- **BUSINESS / FA VERTELT NOOIT HET VOLLEDIGE VERHAAL.**
  - Men ziet een demo en men denkt: “het werkt”.

# De toekomst is altijd beter dan het verleden

## Het verleden is altijd beter dan de toekomst

- **NO WAY.**
  - Je krijgt gewoon andere problemen.
  - De problemen verschuiven (naar een andere functie / team).
- **DE WET VAN BEHOUD VAN ELLENDE.**
  - Op het einde van de rit, komt het probleem bij de tester terecht.
  - Of bij de persoon die software gebruikt.
- **BUSINESS / FA / DEVELOPERS VERTELLEN NOOIT HET VOLLEDIGE VERHAAL.**
  - En soms een verschillende verhaal.

# Wat gebeurt er met de schattingen?

- **BEPALEN VAN HET BUDGET.**
  - Tijd is geld, maar tijd is niet hetzelfde als geld.
- **BESLISSSEN OF IETS DOORGAAT.**
  - Is het de moeite om het te doen? Kosten baten analyse.
- **HET GEVAAR VAN OVERSCHATTEN?**
  - Het zal minder kosten dan gedacht.
  - Men doet iets niet omdat men het te hoog heeft ingeschat.
- **HET GEVAAR VAN ONDERSCHATTEN?**
  - Het zal meer kosten dan gedacht.
  - Er is geen weg terug als men al gestart is. (de euro verkopen)

# Hoe schatten?

Het project in kleinere stukken opsplitsen

Taken aan teamleden assigneren  
(niet altijd op voorhand)

Schatten van de “werklast” per  
taak

Valideren van de schatting

# Een vergelijking met testen opzetten

Analyseren van de requirements + Crosscheck

Denk de testscenario's (cases) uit + Crosscheck

Testscenario's (cases) uitvoeren + Crosscheck

Rapporteer status + Crosscheck

## 6 schattingstechnieken

- **HISTORISCHE GEGEVENS ALS LEERCURVE**
- **LAAT IEMAND ANDERS VOOR JOU SCHATTEN**
- **SCHAT IN KLEINE STUKKEN, LAAT RUIMTE VOOR MARGE**
- **HET DRIE PUNTEN SYSTEEM**
- **BEREKEN HET “VAGE” IN IETS**
- **SCHAT GEDURENDE EEN LAAG PUNT IN JE DAG**

# Historische gegevens als leercurve

- **HOELANG DUURT IETS GEWOONLIJK?**
  - Kijk naar andere firmas / projecten
- **IN PLAATS VAN**
  - Wat zijn onze specifieke kenmerken en moeilijkheden in ons specifiek probleem?
- **EEN VOORBEELD**
  - Gaan winkelen met een lijstje
  - Gaan winkelen met een budget
- **NADEEL**
  - Je hebt niet altijd historische gegevens

# Laat iemand anders voor jou schatten

- **WE ZIJN SLECHT IN SCHATTEN VOOR ONSZELF**
- **WE ZIJN HÉÉL GOED IN SCHATTEN HOELANG IEMAND ANDERS ER OVER DOET OM IETS AF TE WERKEN**
- **VOORBEELD**
  - Het schrijven van het alfabet van A naar Z
  - Het schrijven van het alfabet van Z naar A
  - Het schrijven van het alfabet tegelijkertijd van A naar Z en van Z naar A.

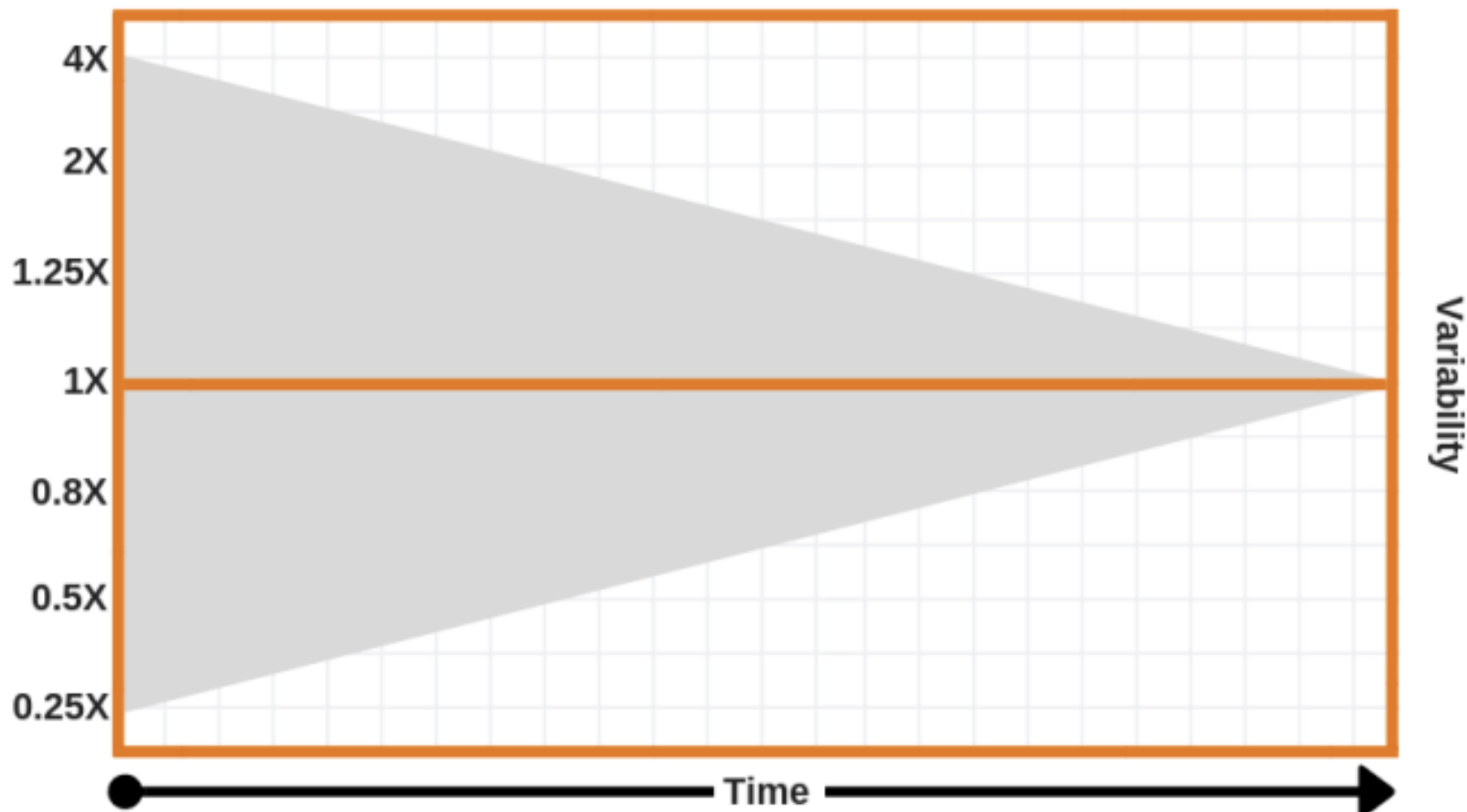


# Schat in kleine stukken, laat ruimte voor marge

- **ER ZIJN BEKENDE BEKENDHEDEN**
  - We weten wat er is
- **ER ZIJN DINGEN WAARVAN WE WETEN DAT WE ZE WETEN**
  - En we zijn er zeker van
- **WE WETEN OOK DAT ER BEKENDE ONBEKENDEN ZIJN**
  - We weten dat er enkele dingen zijn die we niet weten
- **ER ZIJN ONBEKENDE ONBEKENDEN**
  - De dingen waarvan we niet weten dat we ze niet weten

# Schat in kleine stukken, laat ruimte voor marge

- **DE KEGEL VAN ONZEKERHEID IN HET BEGIN**
  - iets wat 2 xx geschat is, kan 8 xx zijn, maar ook  $\frac{1}{2}$  xx



# Het drie punten systeem

- **HET SCHATTEN VAN HET “BESTE-GEVAL” SCENARIO**
  - Wat als alles goed gaat?
  - Dat geeft je een waarde “O” (bijvoorbeeld → 30)
- **HET SCHATTEN VAN HET “SLECHTSTE-GEVAL” SCENARIO**
  - Wat als alles fout gaat?
  - Dat geeft je een waarde “P” (bijvoorbeeld → 55)
- **HET SCHATTEN VAN HET “GEWOONLIJK-GEVAL” SCENARIO**
  - Wat als gewoon verloopt zoals gewoonlijk?
  - Dat geeft je een waarde “ML” (bijvoorbeeld → 40)
- **NEEM HET GEMIDDELDE (O + P + ML) / 3**
  - $(30 + 55 + 40) / 3 \rightarrow 41,67$

# Iets ingewikkelder “Pert Kernel”

- **HET SCHATTEN VAN HET “BESTE-GEVAL” SCENARIO**
  - Dat geeft je een waarde “O” (bijvoorbeeld → 30)
- **HET SCHATTEN VAN HET “SLECHTSTE-GEVAL” SCENARIO**
  - Dat geeft je een waarde “P” (bijvoorbeeld → 55)
- **HET SCHATTEN VAN HET “GEWOONLIJK-GEVAL” SCENARIO**
  - Dat geeft je een waarde “ML” (bijvoorbeeld → 40)
- **$(O + 4 ML + P) / 6$** 
  - $(30 + 4 * 40 + 55) / 6 \rightarrow 40,83$

# Wat is de foutmarge?

- **HET SCHATTEN VAN HET “BESTE-GEVAL” SCENARIO**
  - Dat geeft je een waarde “O” (bijvoorbeeld → 30)
- **HET SCHATTEN VAN HET “SLECHTSTE-GEVAL” SCENARIO**
  - Dat geeft je een waarde “P” (bijvoorbeeld → 55)
- **$(P - O)/6$  (STANDAARDDEVIATIE)**
  - Boven of onder je schatting.
  - $(55 - 30) / 6 \rightarrow 4,16$
- **ACCEPTABELE SCHATTING**
  - $40,83 \pm 4,16 \rightarrow$  Tussen 36,66 en 45

# Bereken het “vage” in iets

- **VERGELIJK DIT MET “VELOCITY” IN SCRUM**
- **NOTEER JE SCHATTINGEN**
- **NOTEER JE EFFECTIEVE TIJDEN**
- **GEEF JE TAKEN EEN GEWICHT**
  - Wat is het gewicht van een taak of user story
- **BEREKEN HET VERSCHIL, EN NEEM DAAR EEN “GEWOGEN” GEMIDDELDE VAN.**
  - $\sum(Verschil * Waarde) / \sum(Waarde)$

# Bereken het “vage” in iets – Een voorbeeld

- **3 TAKEN**

- Taak A, schatting 2, gewicht 1
- Taak B, schatting 2, gewicht 1
- Taak C, schatting 6, gewicht 2
  - Dit wil zeggen dat je wilt dat de belangrijkheid van taak C twee keer zo hoog is als de andere taken

- **NOTEER JE EFFECTIEVE TIJDEN**

- Taak A, effectief 2 → Goede schatting
- Taak B, effectief 3 → 1 teveel
- Taak C, effectief 9 → 3 teveel, maar het weegt iets zwaarder door

- **BEREKENING**

- $(1 * 0 + 1 * 1 + 2 * 3) / (1 + 1 + 2) \rightarrow 7 / 4 \rightarrow 1.75$  (fout geschat)

# Schat gedurende een laag punt in je dag

- **JE DAG HEEFT EEN “BIORITME”**
  - Wanneer ben je positief, negatief, creatief, lusteloos
  - Je kent het gevoel wel, net gegeten, en je wilt in een hangmat liggen.
- **SCHAT NOOIT OP JE PIEKMOMENT**
- **ALTIJD OP JE LAAGTE PUNT**



# Wat nog?

- **MAAK ABSTRACTIE VAN DINGEN EN DETAILS**
  - Accuraat zijn is beter dan precies zijn
- **DE WET VAN DE GROTE GETALLEN**
  - Verschil tussen € 1.000 en € 10.000
  - Verschil tussen € 1.000.000 en € 1.100.000
- **ZANDZAKJES LEGGEN (BUFFEREN VAN TIJD)**
  - Dat is systematisch overschatten.
    - Dit heeft geen enkele zin, je 'liegt' tegen jezelf en het team.

# SOFTWARE TESTEN PLANNEN

*Planning  
1 werkuur*

## Er zijn altijd twee vragen

Hoelang gaat  
het duren?

Hoeveel gaat  
het kosten?

# Wat moet je plannen?

Hulpmiddelen

Tijd

Menselijke capaciteit

Budget en afhankelijkheden  
van alle bovenstaande

# Wat moet je plannen?

## Grootte (punten)

- Complexiteit van de onderdelen

## Tijd

- (tijdspanne + doorlooptijd)

## Kost

## Recheck regelmatig

- Kun je je aan je planning houden?

# Plannen van

- **HULPMIDDELEN**
  - Materiaal, materieel, mensen, ruimte, ...
- **TIJD**
  - Wanneer gaat men het doen?
  - De logische volgorde van dingen
- **MENSELIJKE CAPACITEIT**
  - Wat is de kennis en de ervaring van de mensen?
  - Wanneer zijn die beschikbaar?
- **BUDGET**
  - Hoe en wanneer gaat men het betalen?

# Schattingen die over het budget / tijd gaan

- **WORK BREAKDOWN STRUCTURES**

- Het werk in stukken verdelen
- En maak een MVP
  - Minimal Viable Product
  - Wat heb je minimum nodig om te kunnen starten
    - En meestal is dat het kritische pad van functionaliteiten

- **MEESTAL IS DIT BOTTUM – UP (USER STORY → PRODUCT)**

- Top – Down is meestal op ervaring

- **SOMS IS DAT DE REALITEIT**

- Het kan niet uitgevoerd worden met het beschikbare budget