

SOFTWARE TESTEN – DEFECT MANAGEMENT

Defect Management
2 werkuur

Hoofdtitel: Software Testen

Subtitel: Defect Management

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

Terminologie en hun betekenis

- **SOFTWARE TESTING LIFE CYCLE**
 - STLC
 - Stappen in het testen van de software

- **EEN HOOP VOORBEELDEN**

- **EEN AANTAL METHODOLOGIEËN**

1. **WATERVAL**

2. **V-MODEL**

3. **ITERATIEF MODEL**

Iteratief

4. **SPIRAAL MODEL**

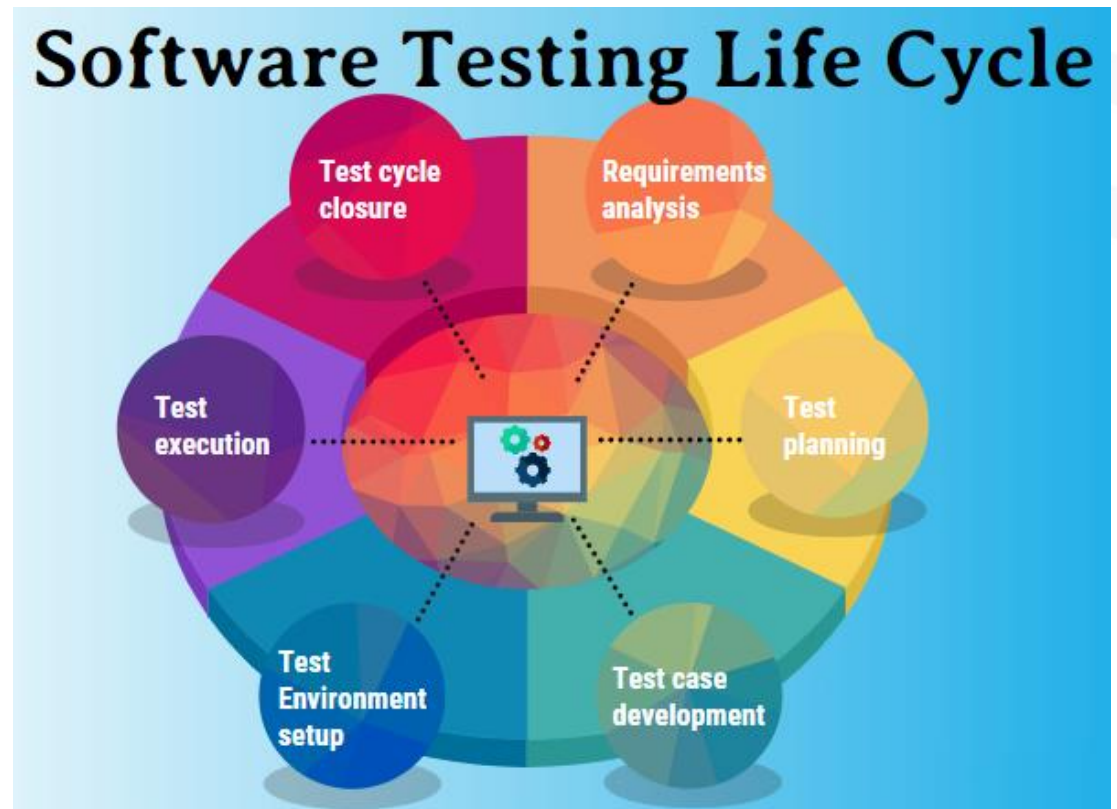
Incrementeel

5. **AGILE / SCRUM**

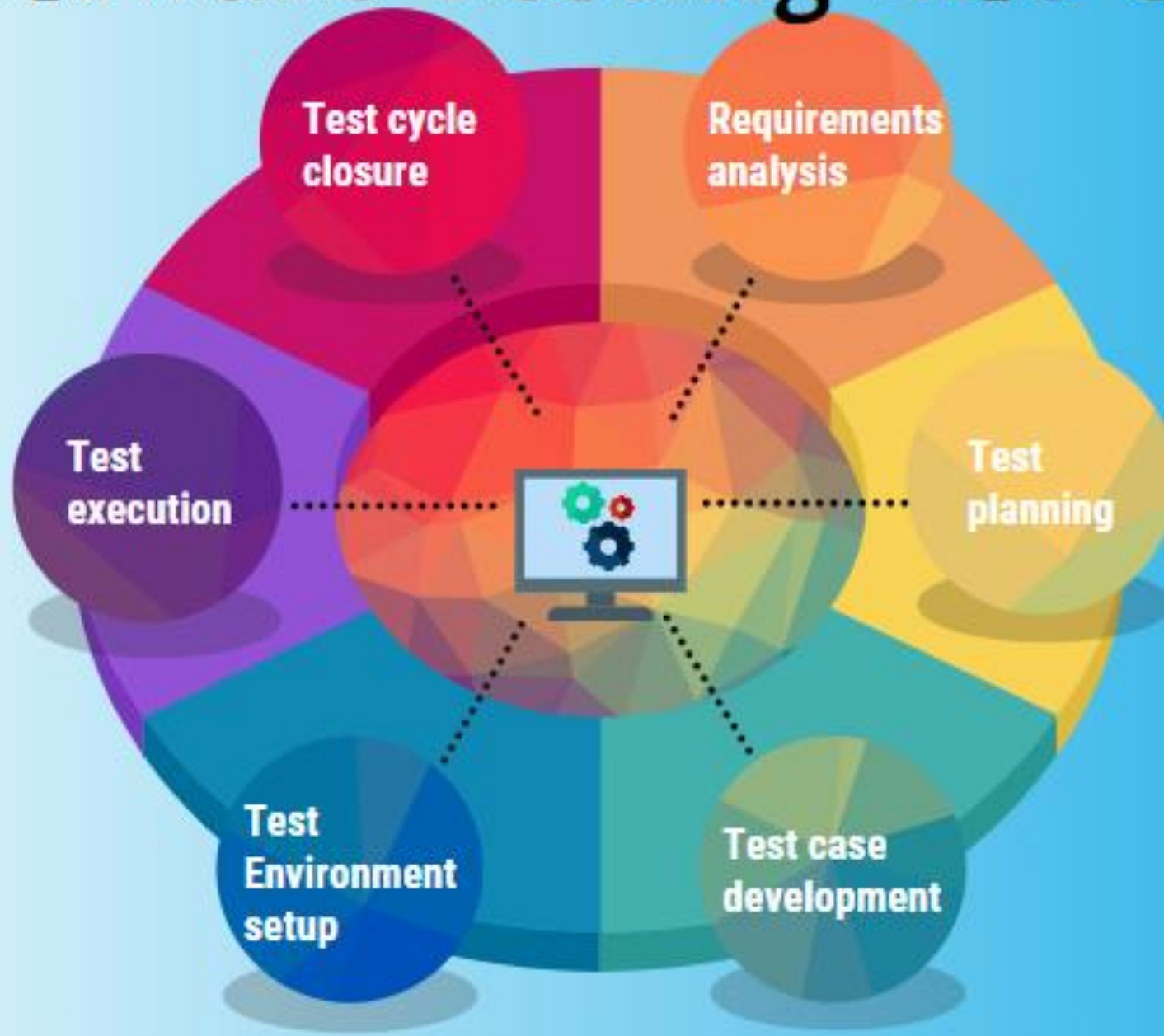
Incrementeel iteratief

6. **DEVOPS**

Het realiseren van incrementeel iteratief



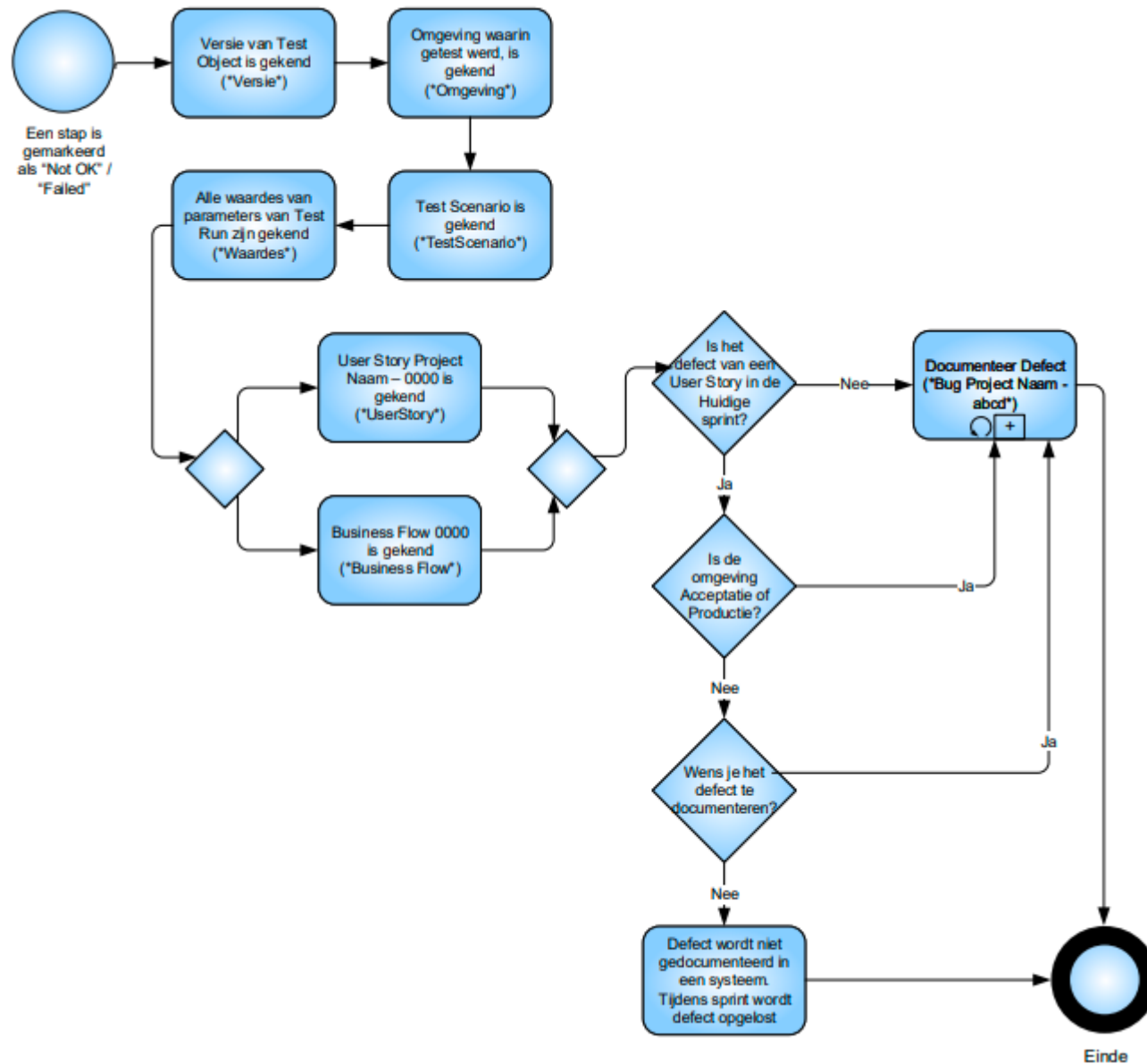
Software Testing Life Cycle



Software Test(ing) Life Cycle

- **JE HEBT DE TOOLING.**
 - Welke software gebruik je om dit te realiseren?
- **JE HEBT HET PROCES.**
 - Hoe ga je met het team om met “issues” (Fouten / Bugs)?

Voorstel tot flow



Stappen in het oplossen van een fout

- **HET ONTDEKKEN VAN EEN FOUT.**
 - Wie, wat, waar, wanneer, hoe?
- **WAT IS DE ERNST VAN DE FOUT?**
- **WAT IS DE PRIORITERING?**
- **WIE GAAT HET OPLOSSEN?**
- **WAT IS DE OPLOSSING?**
- **VERIFIËREN OF DE OPLOSSING GOED IS.**
- **AFSLUITEN.**
- **DE CIJFERS RAPPORTEREN.**

Helpt de tool je met deze dingen?

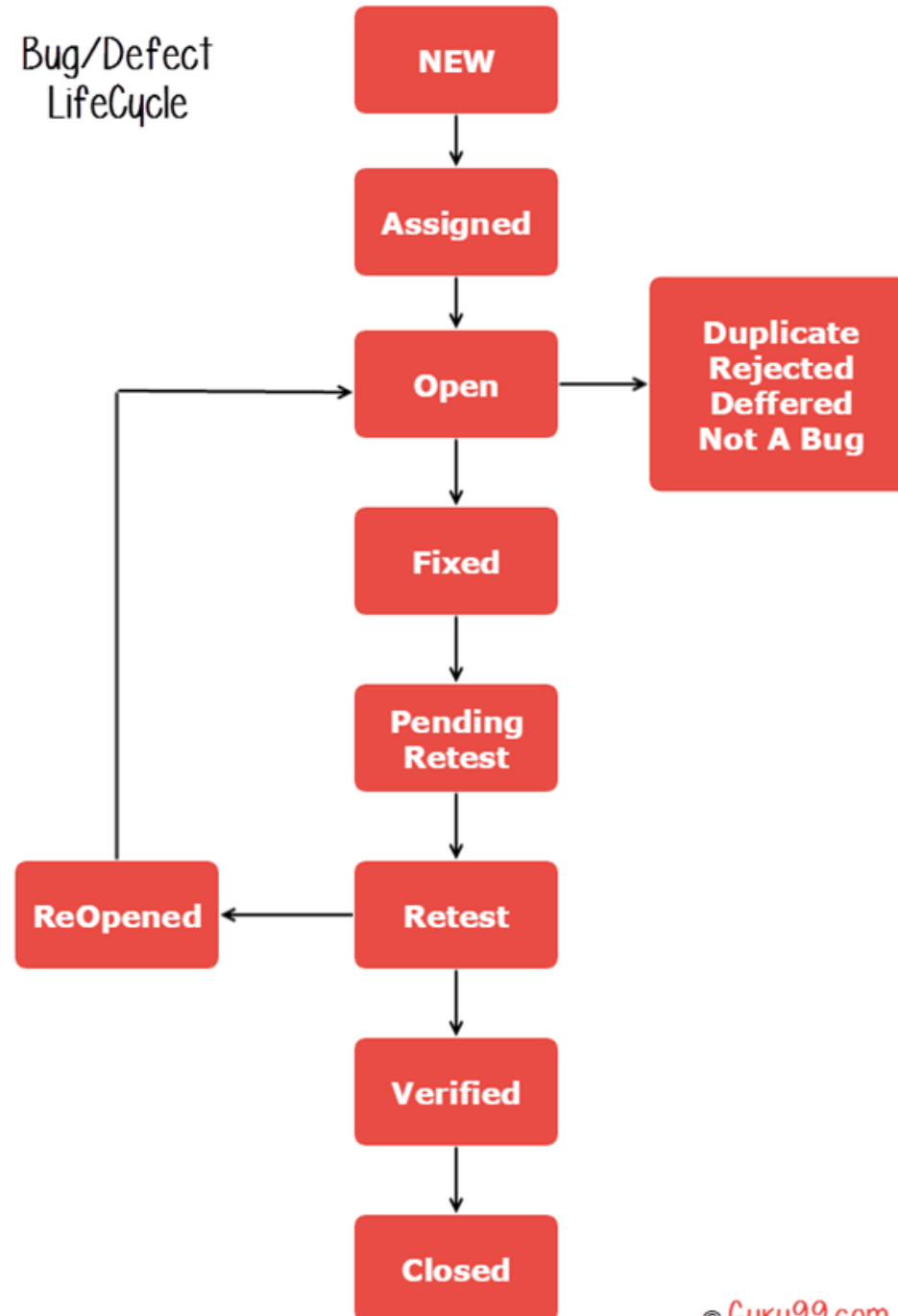
- **HET ONTDEKKEN VAN EEN FOUT.**
 - Wie, wat, waar, wanneer, hoe?
- **WAT IS DE ERNST VAN DE FOUT?**
- **WAT IS DE PRIORITERING?**
- **WIE GAAT HET OPLOSSEN?**
- **WAT IS DE OPLOSSING?**
- **VERIFIËREN OF DE OPLOSSING GOED IS.**
- **AFSLUITEN.**
- **DE CIJFERS RAPPORTEREN.**

Helpt de tool je ook met dit?

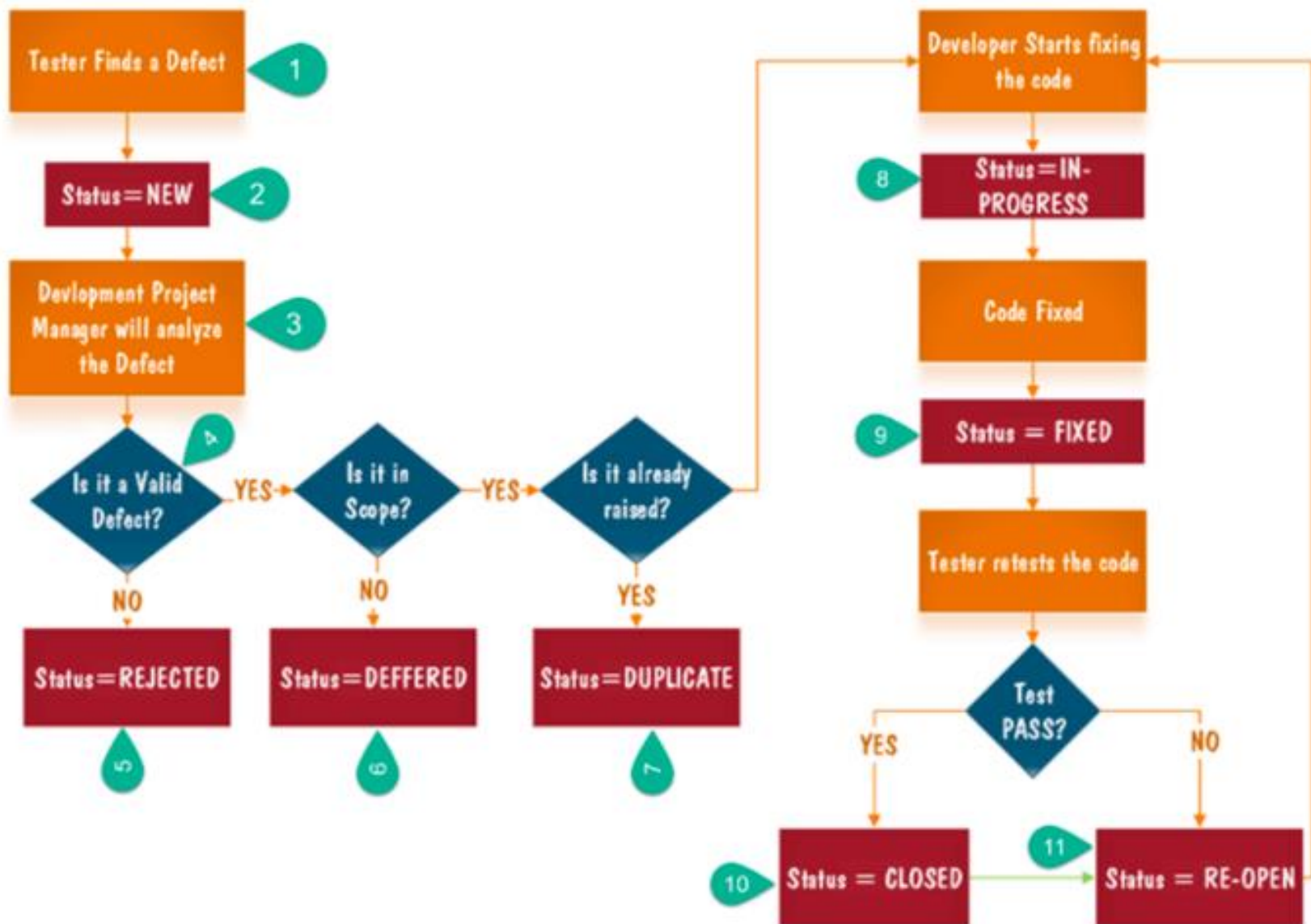
- **HET PROBEREN TE VERMIJDEN VAN SYSTEEM “PARAPLU”.**
- **HET VINDEN VAN FOUTEN IS NIET NEGATIEF.**
 - Het is positief.
 - Namelijk het feit dat je het gevonden hebt.
- **IK HEB NOG NOOIT EEN BIJ EEN PROJECT GEEN FOUTEN GEMAAKT.**
 - Het verhaal van het varken en de kip.
 - Wie is er het meest bij betrokken?

Voorstel tot flow

Bug/Defect
LifeCycle



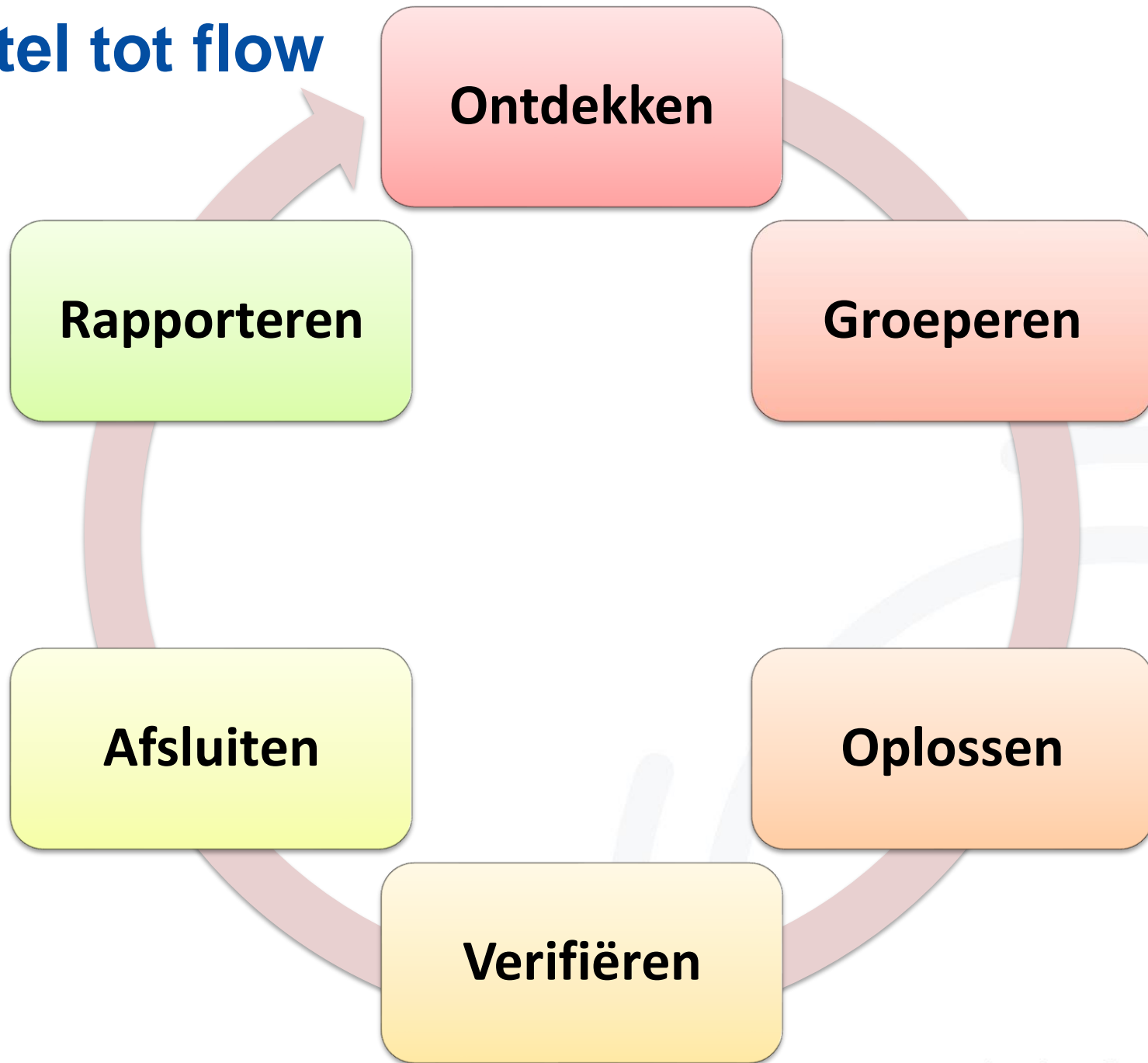
Voorstel tot flow



Herken je deze discussie?

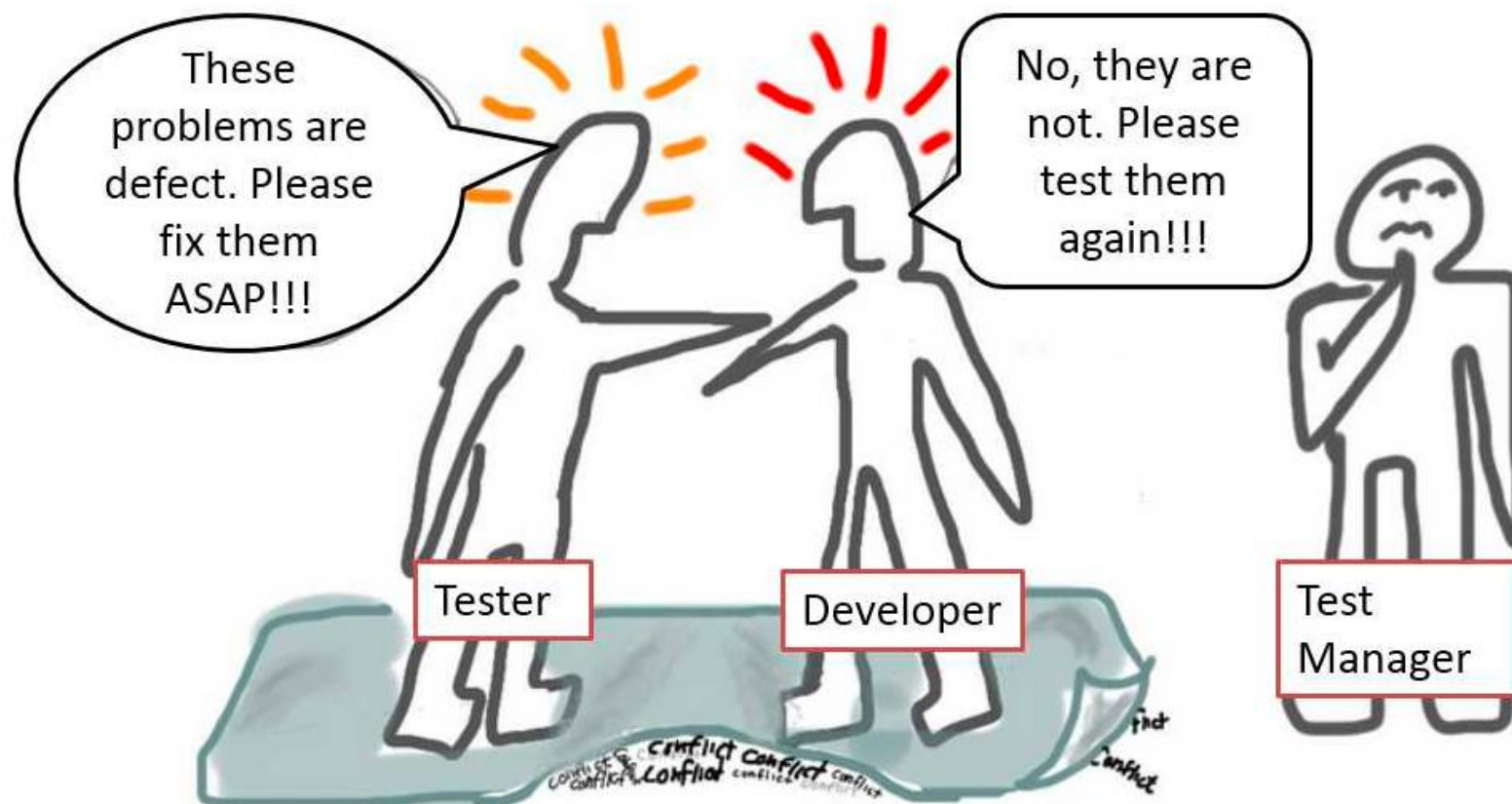
- **EENS IN EEN NIET ZO VER VERLEDEN**
 - Tester: We hebben 24 fouten gevonden.
 - Manager: We verwittigen het ontwikkelteam.
- **EEN WEEK LATER**
 - Ontwikkelaar: We hebben 17 fouten verbeterd.
 - Manager: We verwittigen de testers.
- **NOG EEN WEEK LATER**
 - Tester: We hebben 10 nieuwe fouten gevonden.
 - Manager: We verwittigen het ontwikkelteam.

Voorstel tot flow



Ontdekken

- IEMAND RAPPORTEERT EEN FOUT
- IEMAND KIJKT OF DE FOUT RELEVANT IS



Groeperen

- **WAT IS DE ERNST VAN DE FOUT?**
- **WAT IS DE PRIORITEIT VAN OPLOSSEN?**
- **BEBEIDEN BEPALEN WELKE FOUTEN ER EERST WORDEN OPGELOST.**

Oplossen

- **OFWEL BEHANDEL JE EEN BUG ZOALS EEN USER STORY.**
- **OFWEL NIET 😊.**
- **MIJN ERVARING LEERT DAT BINNEN EEN SPRINT ER NIET TEVEEL TRALALA ROND MOET GEMAAKT WORDEN.**
- **BUITEN EEN SPRINT ZOU IK WEL ALLES DOCUMENTEREN.**
- **VERSCHIL TUSSEN EEN**
 - Fout
 - Change Request

Verifiëren

- **DE TEST WORDT OPNIEUW UITGEVOERD.**
 - Fout kan niet gereproduceerd worden.
 - Fout kan wel gereproduceerd worden.
 - Er gaat iets anders fout.

Afsluiten

- **ALS DE FOUT NIET KAN GEREPRODUCEERD WORDEN**
 - Dan wordt de issue als “Closed” beschouwd.
 - Binnen in de test omgeving.
- **PRODUCTIE FOUTEN WORDEN PAS OP “CLOSED” GEZET**
 - Wanneer de fout ook effectief in productie is opgelost.

Rapporteren

- **GEBRUIK HIERVOOR JE TOOLING.**
- **GEBRUIK DE STANDAARD RAPPORTEN EN WIJK DAAR NIET TEVEEL VAN AF.**
- **PROBEER MANAGEMENT ZELF HUN RAPPORTEN TE LATEN TREKKEN.**
 - Afgekeurde fouten.
 - Aantal gelogde fouten die geen fout waren.
 - Geannuleerde fouten / totaal aantal fouten (in %)
 - Doorgelekte fouten.
 - Fouten in productie die niet gevonden werden in testomgeving.
 - Aantal fouten in productie / totaal aantal fouten (in %)