Onderlegger / Checklist C24 (Crebo 25998)

Kerntaak 1

B1-K1-W1

Stemt opdracht af, plant werkzaamheden en bewaakt de voortgang

Rubrics

1. De opdracht, doelen en planning zijn afgestemd.
2. De kandidaat houdt bij welke werkzaamheden zijn uitgevoerd, welke nog uitgevoerd moeten worden en gaat na of de planning in gevaar komt.
3. De kandidaat signaleert afwijkingen in de doelen en planning, meldt dit en zoekt (in overleg) naar een (tussen)oplossing.

Onderlegger

() nummers verwijzen naar rubics

1. Opdracht is beschreven (1).
2. Doelen zijn beschreven (1).
3. Planning is beschreven (1, 2).
4. Indien er een opdrachtgever is, dan is er ook afgestemd (1).
5. Projectvoortgang is bewaakt (2, 3)
6. Project bijsturing is uitgevoerd en beschreven (3)

Checklist

(nummers verzijzen naar onderlegger en rubic)

- Is er beschreven **wat** er gemaakt moet worden; hoe ziet het **eindproduct** eruit? (O1, R1)
- Is er beschreven **voor wie** het is bedoeld; wie gaan ermee werken? (O1, R1)
- Welke functionaliteiten moeten **minimaal** werken? (O1, R1)
- Zijn **alle gewenste functionaliteiten** beschreven? (O1, R1)
- Welke **technische uitgangspunten** zijn er (techniek, taal, frameworks)? (O1, R1)
- Welk **doel** dient het project? Welk probleem wordt opgelost en hoe wordt gemeten of dat gelukt is? (O2, R1)
- Bevat de planning duidelijke **taakbeschrijvingen**, plus per taak een geschatte **tijd**, **deadline**, **verantwoordelijke** en **resultaatbeschrijving**? (O3, R1)
  - Taken zijn nooit groter dan 1 dag (8 uur). (O3, R1)
  - Er is voor minimaal 40 uur programmeerwerk gepland (per persoon).
- Is van elke taak bijgehouden **wanneer** deze is **begonnen**, door **wie** en **wanneer** deze **klaar** was? (O5, R2)
- Welke **wijzigingen** zijn er in de planning gemaakt tijdens de uitvoering en zijn deze **onderbouwd**? (O5, R2)
  - Waarom is het gewijzigd? (O5, R2)
  - Wat waren de opties en waarom is deze optie gekozen? (O5, R2)
  - Is er overleg geweest en zo ja, met wie? (O4, R3)

B1-K1-W2

Maakt een technisch ontwerp voor software

Rubrics

1. De eisen, wensen en technische randvoorwaarden zijn vertaald naar een passend, eenduidig en volledig (deel)ontwerp.
2. Er is gebruikt gemaakt van relevante of toepasselijke schematechnieken (bijv. activiteitendiagram, klassendiagram, ERD, use case diagram).
3. De gemaakte keuzes in het ontwerp zijn onderbouwd met steekhoudende argumenten die passen bij de praktijkrichting.

Onderlegger

() nummers verwijzen naar rubics

1. Alle functionalieiten zijn beschreven (1)
2. Alle technische randvoorwaarden zijn beschreven (1)
3. Het ontwerp bevat passende en relevante schema's (zoals ERD, activiteitendiagram, use case, flowdiagram) (2)
4. Alle ontwerpkeuzes zijn onderbouwd met (functionele of technische) argumenten (3).

Checklist

(nummers verwijzen naar onderlegger en rubric)

- Zijn **alle** functionele **eisen** en **wensen** compleet beschreven? (O1, R1)
- Zijn alle **technische** randvoorwaarden volledig en correct vastgelegd (bijv. taal, framework, devices, API's, database)? (O2, R1)
- Is **per eis** of wens aantoonbaar aangegeven hoe deze is vertaald naar het **ontwerp**? (O1, R1)
- Zijn **alle** schermen/componenten/modules uitgewerkt in tekst, schets, mockup of wireframe? (O1, R1)
- Zijn de datastromen, interacties en relaties tussen onderdelen duidelijk beschreven (flowchart)? (O1, R1)
- Bevat het ontwerp minimaal **3 relevante** en **juiste** schema's (ERD, activiteitendiagram, use case, flowdiagram)? (O3, R2)
- Zijn de gebruikte benamingen **consistent** tussen schema's, tekst, datamodel en ontwerp? (O1, R1)
- Is bij elke ontwerpkeuze een inhoudelijke en logische **motivatie** gegeven (functioneel of technisch onderbouwd)? (O4, R3)
- Zijn **alternatieve** ontwerpopties overwogen en is beschreven waarom een bepaalde keuze is gemaakt? (O4, R3)
- Is het ontwerp voldoende gedetailleerd zodat een **andere ontwikkelaar** het systeem kan bouwen zonder extra uitleg? (O1, R1/R2)

B1-K1-W3

Realiseert (onderdelen van) software

Rubrics

1. Er is voldoende inhoud van de functionaliteiten gerealiseerd binnen de gestelde/geplande tijd.
2. De opgeleverde functionaliteiten voldoen aan de eisen en wensen.
3. De kwaliteit van de code is goed.
4. Versiebeheer is effectief toegepast.

Onderlegger

() nummers verwijzen naar rubics

1. De geplande functionaliteiten zijn binnen de afgesproken tijd gerealiseerd of er is duidelijk verantwoord waarom dit niet volledig gelukt is (1).
2. Er wordt ten minste 40 uur verantwoord voor dit werkproces. (1)
3. De gerealiseerde functionaliteiten werken volgens de vooraf opgestelde eisen, wensen en acceptatiecriteria (2).
4. Edge cases, foutafhandeling en uitzonderingssituaties zijn functioneel juist afgehandeld (2).
5. De code heeft een duidelijke structuur (3)
6. De code volgt relevante best practices (bijv. DRY, SRP, scheiding van concerns, herbruikbare functies, valide input-checks) (3).
7. De opmaak van de code is netjes (inspringen). (3)
8. Er is voldoende zinvol commentaar toegevoegd. (3)
9. De naamgeving van variabelen, functies, methods en classes is consistent en duidelijk (3)
10. De kandidaat heeft commit-historie die laat zien dat er regelmatig, logisch en stapsgewijs gewerkt is (4).
11. Commits bevatten betekenisvolle commit messages die duidelijk maken wát er is veranderd en waarom (4).
12. De kandidaat gebruikt branches, merges en/of pull requests op een manier die past bij de omvang van het project (4).

Checklist

(nummers verwijzen naar onderlegger en rubric)

- Zijn de geplande functionaliteiten gerealiseerd binnen de afgesproken tijd, of is duidelijk verantwoord waarom dit niet gelukt is? (O1, R1)
- Is er **minimaal 40 uur** aan werkzaamheden verantwoord binnen dit werkproces? (O2, R1)
- Werken **alle** opgeleverde functionaliteiten volgens de vooraf opgestelde eisen en wensen? (O3, R2)
- Zijn **edge cases**, foutafhandeling en uitzonderingssituaties correct en functioneel afgehandeld? (O4, R2)
- Is de code logisch gestructureerd en duidelijk opgezet waarbij de code is opgedeeld in een duidelijke **directory-structuur** en is de code zinvol is **opgedeeld** in **bestanden**? (O5, R3)
- Volgt de code relevante **best practices** (DRY, SRP, scheiding van concerns, herbruikbare functies, inputvalidatie)? (O6, R3)
- Is de code **netjes** opgemaakt, inclusief correct **inspringen**? (O7, R3)
- Is er voldoende **zinvol commentaar** toegevoegd waar dat nodig is? (O8, R3)
- Zijn **namen** van o.m. variabelen, functies, methods en classes **consistent** en **duidelijk** gekozen? (O9, R3)
- Laat de **commit-historie** zien dat er **regelmatig**, logisch en stapsgewijs gewerkt is? (O10, R4)
- Bevatten commits betekenisvolle **commit messages** die duidelijk maken wát en waarom iets is aangepast? (O11, R4)

Extra toeliching

DRY (Don’t Repeat Yourself)
Voorkomt dubbele code. Wanneer dezelfde logica op meerdere plekken staat, breng je het onder in één gedeelde functie of module. Dit maakt onderhoud eenvoudiger en vermindert fouten.

Herbruikbare functies
Functies die zo zijn geschreven dat ze op meerdere plekken en in verschillende situaties toepasbaar zijn. Ze zijn klein, duidelijk gedefinieerd en afhankelijk van zo min mogelijk specifieke context. Deze functies zijn de bouwblokken van de code.

SRP (Single Responsibility Principle)
Elke functie, klasse of module moet precies één duidelijke taak hebben. Dit maakt code begrijpelijker, testbaarder en minder foutgevoelig.

Scheiding van concerns
Splits verschillende verantwoordelijkheden in aparte lagen of bestanden. Bijvoorbeeld: HTML voor structuur, CSS voor vormgeving, JavaScript/PHP voor logica. Dit verhoogt overzicht en

onderhoudbaarheid.

Inputvalidatie
Controleer alle gegevens die van gebruikers, API’s of formulieren binnenkomen. Je controleert of ze aanwezig, geldig en veilig zijn voordat je ze gebruikt, om fouten, beveiligingslekken en onverwacht gedrag te voorkomen.

Abstractie
Abstractie verbergt details, waardoor functies herbruikbaar worden en duplicatie verdwijnt.

B1-K1-W4

Test software

Rubrics

1. De kandidaat heeft passende testvormen en testmethodieken gekozen.
2. De kandidaat heeft voor alle toegewezen of geplande functionaliteiten testscenario’s of testcases gemaakt.
3. Het testrapport bevat testresultaten van alle functionaliteiten. Alle resultaten worden voorzien van de juiste conclusies.

Onderlegger

() nummers verwijzen naar rubics

1. Je voert ten minste één integratietest uit. (1)
2. Er is minimaal één geautomatiseerde test uitgevoerd met een geschikte tool (bijv. Selenium, Playwright of een alternatieve testtool). (1)
3. Er is risk-based getest: op basis van een risicoanalyse zijn de belangrijkste tests als eerste uitgevoerd. (1)
4. Voor alle toegewezen of geplande functionaliteiten zijn testscenario’s of testcases opgesteld met duidelijke input, stappen en verwachte output. (2)
5. De testcases bevatten zowel happy flows, edge cases als foutgevallen. (2)
6. Je bepaalt waar performance een risico kan vormen en voert indien relevant performance-gerelateerde tests uit. (1/2 – optioneel afhankelijk van opdracht)
7. Testscenario’s zijn volledig en zo beschreven dat een andere tester ze onafhankelijk kan uitvoeren. (2)

8. Het testrapport bevat de resultaten van alle uitgevoerde tests, inclusief datum, testomgeving en eventuele bijzonderheden. (3)
9. Bij elk testresultaat is duidelijk aangegeven of de test geslaagd is, inclusief een toelichting bij afwijkingen. (3)
10. Er zijn juiste en logische conclusies getrokken op basis van de testresultaten, inclusief aanbevelingen of vervolgstappen. (3)

Checklist

(nummers verwijzen naar onderlegger en rubric)

- Is er ten minste één **integratietest** uitgevoerd? (O1, R1)
- Is er minimaal één **geautomatiseerde test** uitgevoerd met een geschikte tool (bijv. Selenium, Playwright)? (O2, R1)
- Is er risk-based getest op basis van een vooraf uitgevoerde **risicoanalyse**? (O3, R1)
- Zijn er voor **alle** functionaliteiten testscenario's of testcases opgesteld? (O4, R2)
- Bevatten de testcases **happy flows**, **edge cases** en **foutgevallen**? (O5, R2)
- Zijn **performance**-gerelateerde tests uitgevoerd? (O6, R1/R2) - *moet dit?*
- Zijn de testscenario's **volledig** en zo beschreven dat een andere tester ze **onafhankelijk** kan uitvoeren? (O7, R2)
- Bevat het testrapport **resultaten** van alle uitgevoerde tests, inclusief datum, omgeving en bijzonderheden? (O8, R3)
- Is bij elk testresultaat duidelijk aangegeven of de test **geslaagd** is en is er een **toelichting** bij afwijkingen? (O9, R3)
- Zijn er **logische** en **correcte conclusies** getrokken op basis van alle testresultaten, inclusief **aanbevelingen** of vervolgacties? (O10, R3)

B1-K1-W5

Doet verbetervoorstellen voor de software

Rubrics

De kandidaat analyseert systematisch alle beschikbare informatiebronnen voor mogelijke aanpassingen aan de software.

De kandidaat interpreteert en vertaalt wensen, reacties, testresultaten en/of meldingen naar realiseerbare verbetervoorstellen.

De kandidaat stelt vast welke werkzaamheden benodigd zijn, evenals een haalbare planning.

Onderlegger

() nummers verwijzen naar rubics

1. Alle relevante informatiebronnen zijn systematisch verzameld en geanalyseerd (bijv. testresultaten, foutmeldingen, logs, gebruikersfeedback, codekwaliteit, performancegegevens). (1)
2. De kandidaat toont aan dat de informatieobjectief en volledig is geïnterpreteerd (bijv. clustering van problemen, oorzaakanalyse, categorisatie). (1)
3. Er is een duidelijke probleemanalyse gemaakt, inclusief onderliggende oorzaken (root cause analysis). (1)
4. Wensen, reacties, testresultaten of meldingen zijn vertaald naar één of meerdere concrete, haalbare verbetervoorstellen. (2)
5. De verbetervoorstellen zijn functioneel en technisch onderbouwd, inclusief motivatie waarom deze verbeteringen waardevol zijn. (2)
6. Elk verbetervoorstel bevat een duidelijke beschrijving van de impact en gevolgen voor gebruikers, data, functionaliteit of onderhoud. (2)
7. Er is onderscheid gemaakt tussen kleine verbeteringen en grotere wijzigingen (bijv. quick wins vs. major improvements). (2)
8. Er is per verbetervoorstel vastgesteld welke werkzaamheden nodig zijn (bijv. analyse, ontwerp, codewijziging, testen, documentatie). (3)
9. De benodigde tijd, afhankelijkheden en risico’s zijn ingeschat en beschreven. (3)
10. Er is een haalbare planning opgesteld voor de realisatie van de verbeteringen (bijv. volgorde, prioriteit, tijdsinschatting, sprintplanning). (3)

Checklist

(nummers verwijzen naar onderlegger en rubric)

1. Zijn alle relevante informatiebronnen verzameld en systematisch geanalyseerd (zoals logs, testresultaten, foutmeldingen, gebruikersfeedback)? (O1, R1)
2. Is de verzamelde informatie objectief en volledig geïnterpreteerd (bijv. clustering, patronen, oorzaken)? (O2, R1)
3. Is er een duidelijke probleemanalyse gemaakt, inclusief onderliggende oorzaken (root cause analysis)? (O3, R1)
4. Zijn wensen, reacties, testresultaten of meldingen vertaald naar concrete, haalbare verbetervoorstellen? (O4, R2)

5. Zijn de verbetervoorstellen functioneel en technisch onderbouwd? (O5, R2)
6. Bevat elk verbetervoorstel een duidelijke beschrijving van de impact (gebruikers, data, functionaliteit, onderhoud)? (O6, R2)
7. Is onderscheid gemaakt tussen kleine verbeteringen (quick wins) en grotere wijzigingen? (O7, R2)
8. Is per verbetervoorstel vastgesteld welke werkzaamheden nodig zijn (ontwerp, code, testen, documentatie)? (O8, R3)
9. Zijn tijdsinschatting, afhankelijkheden en risico’s beschreven? (O9, R3)
10. Is een haalbare planning opgesteld voor de uitvoering van de verbeteringen (volgorde, prioriteit, tijdsinschatting)? (O10, R3)

Revision #8
Created 2025-11-19 09:51:35 UTC by Max
Updated 2025-12-18 09:22:12 UTC by Max