

Detailtestplan Softwarerealisatie

**{Projectnaam}**

Rubriceringsniveau {Rubriceringsniveau}

Versie {Versienummer}, {Datum}



Inhoudsopgave

##### Colofon

###### Rubricering

Rubricering conform [VIRBI 2013, art. 4](https://wetten.overheid.nl/BWBR0033507/2013-06-01#Artikel4).

{Verwijder deze paragraaf en de rubricering op de titelpagina als rubricering niet van toepassing is}

|  |  |  |
| --- | --- | --- |
| Rubriceringsniveau | Rubriceringsduur | Vaststeller |
| {Rubriceringsniveau} | {Rubriceringsduur} | {Vaststeller van de rubricering: minister, staatssecretaris, secretaris-generaal of een door de secretaris-generaal aangewezen rubriceringsambtenaar} |

###### Goedkeuring

|  |  |  |
| --- | --- | --- |
| Versie | Datum goedkeuring | Goedgekeurd door |
| {versie} | {datum} | {naam} |

###### Revisiehistorie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Versie | Datum | Status | Auteur | Reviewers | Opmerkingen |
| {versie} | {datum} | {status} | {naam} | {namen} | {opmerkingen} |

###### Betrokkenen bij dit document

{Neem in onderstaande tabel de auteurs, reviewers en goedkeurders van dit document op}

|  |  |  |
| --- | --- | --- |
| Organisatie | Functie/rol | Naam |
| {opdrachtgevende organisatie} | Opdrachtgever | {naam} |
| {opdrachtgevende organisatie} | Projectleider | {naam} |
| {opdrachtgevende organisatie} | Product owner | {naam} |
| {opdrachtgevende organisatie} | Kwaliteitsmanager | {naam} |
| {beheerorganisatie} | Projectleider | {naam} |
| {beheerorganisatie} | Kwaliteitsmanager | {naam} |
| ICTU | Projectleider | {naam} |
| ICTU | Software delivery manager | {naam} |
| ICTU | Kwaliteitsmanager | {naam} |

###### Template versie

Versie 4.0.4, 20-12-2024

Verbeterpunten t.a.v. deze template graag melden via [GitHub](https://github.com/ICTU/Kwaliteitsaanpak/issues).

# Managementsamenvatting

{Managementsamenvatting}

# Inleiding

## Over dit document

Dit detailtestplan beschrijft welke testen en testsoorten ICTU gebruikt om de kwaliteit van de projectresultaten te testen gedurende de realisatiefase van het project. Uitgangspunten voor dit testplan zijn de ICTU Kwaliteitsaanpak Softwareontwikkeling en het Mastertestplan.

## Doelgroep

Dit document is bedoeld voor iedereen die betrokken is bij het testtraject van {het project}.

## Kaders

De volgende kaders zijn van toepassing op het projectresultaat:

|  |  |
| --- | --- |
| Volgnummer | Kader |
| K01 | NEN-ISO/IEC 27001:2017 en NEN-ISO/IEC 27002:2017, VIR 2007, VIRBI 2013 en BIO voor het inrichten en beheren van informatiebeveiliging in brede zin. |
| K02 | NCSC ICT-beveiligingsrichtlijnen voor webapplicaties |
| K03 | OWASP Top-10 |
| K04 | ISO 9241-210:2019 Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems |
| K05 | WCAG2.2 (Web Content Accessibility Guidelines) voor eisen met betrekking tot toegankelijkheid |
| K06 | NEN-ISO/IEC 25010:2011 voor het specificeren van productkwaliteit |
| K07 | NORA - Referentiearchitectuur voor de Nederlandse Overheid |

## Uitgangspunten

De volgende uitgangspunten zijn van toepassing op dit document:

|  |  |
| --- | --- |
| Volgnummer | Uitgangspunt |
| U01 | De realisatie van de software wordt door ICTU uitgevoerd. Het beheer wordt door {beheerorganisatie} uitgevoerd. |
| U02 | De als testbasis geïdentificeerde documenten dienen door alle acceptanten, inclusief het testteam, te zijn geaccordeerd, alvorens met de testspecificatie kan worden begonnen. |
| U03 | Per release of per sprint kan de product owner besluiten om bepaalde functionaliteit, bijvoorbeeld geleverd door externe partijen, niet te testen. Indien dit voorkomt, dan zal dit expliciet worden opgenomen in de managementsamenvatting van het vrijgaveadvies. |
| U04 | De multidisciplinaire samenstelling van de Scrumteams — test engineers en ontwikkelaars zitten in hetzelfde team — geeft mogelijkheid tot snelle interactie. Issues gevonden binnen een sprint kunnen hierdoor vaak nog binnen dezelfde sprint worden opgelost en hoeven dus niet apart geadministreerd te worden. |
| U05 | Issues gevonden tijdens de acceptatietesten of in productie worden op de backlog verwerkt samen met de user stories. |
| U06 | De ontwikkel- en testplanning zijn met elkaar verweven. Het ontwikkelen en testen van de user stories zijn onlosmakelijk met elkaar verbonden en beginnen op hetzelfde moment. |
| U07 | De testomgeving wordt conform planning tijdig en correct werkend opgeleverd. |
| U08 | Er is voldoende en gepaste testdata ter beschikking. Zie de paragraaf Testdata in het hoofdstuk Infrastructuur voor een beschrijving van de testdata. |
| U09 | De testspecificatie voor een testsoort start pas als aan de entry-criteria (Definition of Ready), die hiervoor gelden, is voldaan. Hieraan moet voldaan zijn voor het deel (van de testbasis) dat in de betreffende sprint gerealiseerd en getest wordt (bijvoorbeeld een aantal bij elkaar horende user stories). |
| U10 | De gevonden en openstaande bevindingen worden structureel besproken in het bevindingenoverleg. Als er veel bevindingen zijn of meer ernstige bevindingen dan verwacht, dan wordt bijgestuurd op de kwaliteit van het ontwikkeltraject. Dat heeft ook impact op de planning en begroting. |
| U11 | Nieuwe functionaliteit wordt vastgelegd in de geautomatiseerde regressietest. |
| U12 | {Indien noodzakelijk vindt er een end-to-end test-plaats in samenwerking met betrokkenen bij de gekoppelde systemen.} |
| {volgnummer} | {uitgangspunt} |

## Relatie met andere documenten

Tijdens de voorfase wordt in diverse documenten beschreven welke eisen en wensen aan de oplossing gesteld zijn en hoe de oplossing werkt. Dit detailtestplan beschrijft hoe deze eisen en wensen getest worden.

De volgende documenten vormen de basis voor dit detailtestplan:

* Mastertestplan {documentreferentie},
* Projectplan {documentreferentie},
* Softwarearchitectuurdocument {documentreferentie},
* Kwaliteitsplan {documentreferentie},
* Informatiebeveiligingsplan {documentreferentie}.

De testbasis wordt gevormd door de volgende documenten:

* Globaal functioneel ontwerp {documentreferentie},
* Niet-functionele eisen {documentreferentie}.

## Leeswijzer

Hoofdstuk 2 beschrijft de scope van dit testplan. Hoofdstuk 3 beschrijft de testaanpak en de verschillende soorten testen die in het project toegepast worden. Hoofdstuk 4 beschrijft de infrastructuur voor het testen, de te gebruiken testtooling en de testdata. Hoofdstuk 5 beschrijft de testrapportages en testartifacten die geproduceerd zullen worden.

Bijlage A bevat afkortingen en termen die voorkomen in de ICTU Kwaliteitsaanpak Softwareontwikkeling en bijbehorende templates. Bijlage B verwijst naar regelmatig gebruikte bronnen. Bijlage C bevat een beknopte samenvatting van de ICTU Kwaliteitsaanpak Softwareontwikkeling.

# Scope

Dit hoofdstuk beschrijft welke projecten, systemen, componenten, releases, use cases, user stories en niet-functionele eigenschappen binnen of juist buiten de scope van dit detailtestplan vallen.

## Binnen scope

{Beschrijf hier het beschouwingsgebied in gedetailleerde termen als projecten, systemen, releases, versies en user stories die binnen de opdracht vallen.}

## Buiten scope

{Beschrijf hier relevante projecten, systemen, releases, versies en user stories die juist buitne de opdracht vallen.}

# Testaanpak

Dit hoofdstuk beschrijft hoe de testen conform de teststrategie, beschreven in het Mastertestplan, concreet worden aangepakt.

{De Kwaliteitsaanpak schrijft voor dat er in de voorfase een Mastertestplan (door de opdrachtgevende organisatie) is opgesteld. Mocht dit niet zijn opgesteld, dan zal dit detailtestplan minimaal aan de door ICTU gestelde kwaliteitsnormen moeten voldoen.}

## Soorten testen

{Geef hieronder alleen de testsoorten weer die door het project worden uitgevoerd. Zie het kwaliteitsplan voor normen wat betreft de dekking van de verschillende testsoorten.}

Binnen het project worden door ICTU de volgende testsoorten onderscheiden en toegepast:

* Unit testen: De unit testen (op code) worden uitgevoerd door de ontwikkelaars.
* Functionele testen:
* **Smoke test:** Dit is een snelle geautomatiseerde test met weinig diepgang. Deze test wordt uitgevoerd om een build te valideren. Fouten tijdens deze test worden aangemerkt als bouwfouten.
* **Geautomatiseerde regressietest (ART):** Dit is een uitputtende geautomatiseerde test die aantoont dat de bestaande, niet aangepaste functionaliteit nog steeds juist werkt.
* **Handmatig testen van nieuwe functionaliteit:** Het handmatig uitvoeren van fysieke testgevallen om de werking van de nieuwgebouwde functionaliteit te testen.
* **Handmatig regressietest:** Het handmatig uitvoeren van fysieke testgevallen om de werking van de bestaande functionaliteit te controleren. Deze testgevallen zijn veelal te complex om te automatiseren.
* Niet-functionele testen:
* **Performancetesten:** Het testen van de snelheid van afhandeling van bepaalde functies van het systeem onder een vooraf gedefinieerde belasting. Performancetesten vinden bij voorkeur plaats in een productie-like omgeving, maar kunnen ook in een niet-productie-like omgeving plaatsvinden ten behoeve van het volgen van de relatieve performance van verschillende versies van de software. Er vinden zowel een loadtest (normale en piekbelasting), als een duurtest (normale belasting voor langere tijd), als een stresstest (verhogen van de belasting totdat het systeem het begeeft) plaats. De Kwaliteitsaanpak schrijft voor dat er tijdens de realisatiefase performancetesten worden uitgevoerd. Deze worden bij voorkeur automatisch uitgevoerd. Belangrijk is dat de performancetest die op de testomgeving wordt uitgevoerd, niet vanzelfsprekend representatief is voor de productieomgeving. Dit betekent dat een opdrachtgevende organisatie op de eigen productieomgeving een performancetest moet (laten) uitvoeren om te controleren dat er aan de gestelde performance-eisen is voldaan.
* **Securitytesten:** Security- en penetratietesten uitgevoerd door een externe partij. Normaliter worden deze minimaal twee maal per jaar of met elke grote release uitgevoerd en niet elke sprint. Securitytesten vinden bij voorkeur plaats in een productie-like omgeving, maar kunnen ook in een niet-productie-like omgeving plaatsvinden ten behoeve van het testen van de beveiliging van de software zelf. De securitytest is inclusief een review van de broncode. Tijdens de realisatie draaien standaard al de volgende securitytesttools mee in de geautomatiseerde pijplijn: SonarQube, OWASP Dependency-Check en/of Dependency-Track, ZAP by Checkmarx en OpenVAS; de bevindingen die uit deze tools komen worden meteen tijdens de realisatie van het systeem opgepakt.
* **Integratietesten:** Tijdens deze test wordt de onderlinge verwerkingswijze tussen de verschillende applicaties getest. Denk hierbij aan gewijzigde applicaties die samen werken met ongewijzigde applicaties. Indien van toepassing zullen hier ook externe systemen bij betrokken worden, in de vorm van stubs. Integratietesten zijn normaal gesproken geautomatiseerde tests. Als onderdeel van de integratietesten wordt getest of de software kan omgaan met fouten in andere applicaties en na een herstart goed blijft functioneren.
* **Gebruikersacceptatietest (GAT):** In tegenstelling tot de ‘traditionele’ watervalmethode biedt agile ontwikkelen meer ruimte voor de gebruiker om te participeren in het ontwikkeltraject. Tijdens elke sprint wordt nieuwe functionaliteit gedemonstreerd door het Scrumteam in een demo-omgeving. {opdrachtgevende organisatie} en/of beheerorganisatie kan een GAT-testomgeving beschikbaar stellen waar gebruikers kunnen werken met de nieuwe applicaties. Bevindingen worden tijdens trainingen of workshops verzameld om in de backlogs verwerkt te worden. De product owner prioriteert vervolgens deze bevindingen.
* **Usabilitytesten:** Het doel van deze test is om te bepalen hoe gemakkelijk / toegankelijk het systeem is in het gebruik ervan. Onderdeel van deze test is de toegankelijkheidstest; hiermee wordt bepaald in welke mate de software voldoet aan de wettelijke vereisten van de Web Content Accessibility Guidelines (WCAG2.2) en eventuele aanvullende toegankelijkheidseisen. Deze toegankelijkheidstesten worden waar mogelijk geautomatiseerd uitgevoerd. De toegankelijkheidseisen die niet geautomatiseerd getest kunnen worden, worden periodiek handmatig getest.

## Agile werkwijze

Het project zal volgens de Scrummethode werken. Dat betekent dat de expertise van de diverse componenten in het landschap bij elkaar zit en dat het testen van de gerealiseerde oplossing in principe tijdens de sprint plaatsvindt.

{Mogelijke uitzonderingen hierop zijn de gebruikersacceptatietest (GAT), securitytest, migratietest (MT), performancetest, ketentest en productacceptatietest (PAT). Deze behoren niet vanzelfsprekend tot de vaste sprintactiviteiten.}

## Testactiviteiten tijdens sprints

Deze paragraaf beschrijft de testactiviteiten die plaatsvinden binnen de grenzen van een sprint. De testengineers nemen actief deel aan de sprintplanning. De stappen voor het testen van een user story zijn als volgt:

1. Ontwikkelen testgevallen in Jira, gebaseerd op user story
	1. Logische testgevallen laten reviewen door teamgenoot;
	2. Reviewcommentaar verwerken.
2. Logische testgevallen uitwerken in fysieke testgevallen
	1. Handmatig uitvoeren op de testomgeving;
	2. Alle voorkomende issues direct oplossen.
3. ART uitvoeren
	1. Alle voorkomende regressie direct oplossen.
4. Na uitvoer fysieke testgevallen en oplossen issues de testgevallen opnemen in de ART.
5. Vaststellen van ART
	1. De laatste nieuwe versie van de ART uitvoeren op de regressie- en de integratietestomgeving;
	2. Eventuele bevindingen en issues dienen direct te worden opgelost en te worden gevolgd door het opnieuw uitvoeren van de ART.

## Entry- en exitcriteria

Dit hoofdstuk beschrijft wat nodig is om te kunnen testen en wanneer er voldoende getest is.

### Entry-criteria

{Benoem hier de criteria waaraan voldaan moet worden voordat het testen kan starten. Voorbeelden: "De testomgeving(en) zijn beschikbaar en ingericht conform de gestelde eisen van de tester" en "Het systeem met omgeving is met goed gevolg door de intake gekomen".}

### Exitcriteria

Aan het eind van de sprint zal er alleen functionaliteit overgedragen worden die voldoet aan de Definition of Done (zie Kwaliteitsplan).

{Benoem hier eventuele specifieke zaken die gerelateerd zijn aan het testen en niet al onderdeel uitmaken van de Definition of Done.}

# Infrastructuur

## Testomgevingen

{Neem hieronder alleen testomgevingen op die voor het project van toepassing zijn.}

Conform BIO-12.1.4 zijn productieomgevingen gescheiden van testomgevingen en wordt er niet getest in productieomgevingen. De onderstaande testomgevingen zijn inzetbaar voor het project:

* **Ontwikkelomgeving:** Op deze omgeving wordt de smoketest uitgevoerd;
* **Testomgeving:** Handmatig testen van nieuwe functionaliteit en het automatiseren van handmatig uitgevoerde testen om op te nemen in de ART;
* **Regressietestomgeving:** ART uitvoeren op omgeving met functionaliteit, zoals meest recent vrijgegeven;
* **Integratietestomgeving:** Alle laatste versies zijn hier uitgerold op een omgeving waar gebruik wordt gemaakt van productie-like data;
* **Performancetestomgeving:** Omgeving gebruikt voor performancetesten;
* **Securitytestomgeving:** Gebruikt voor security- en penetratietesten;
* **Middleware-omgevingen:** Deze testomgevingen worden gebruikt om besturingssystemen, databases, patches en andere 3rd party componenten te testen voor uitrol naar productie;
* **GAT-omgeving:** Op deze omgeving worden de gebruikersacceptatietesten uitgevoerd, inclusief usabilitytestsen.

ICTU beschikt over een private cloud waarin deze omgevingen allemaal kunnen worden ingericht. In welke mate dit productie-like kan, is afhankelijk van de karakteristieken van de productie-omgeving en het beschikbare budget.

## Testtools

{Geef alleen de tools weer die voor het project van toepassing zijn en vul aan waar nodig.}

Tijdens het testen worden de onderstaande tools gebruikt:

* **Axe:** Axe is een testtool voor de Web Content Accessibility Guidelines.
* **Selenium:** Web UI driver library, wordt gebruikt om de browser aan te sturen vanuit de ART.
* **Jenkins:** Jenkins is een Continuous Integration (CI) server die wordt gebruikt om de ART uit te voeren.
* {Andere tools}

## Testdata

Conform BIO-14.3 wordt er niet met productiedata getest. In de ontwikkel-, test- en regressietestomgevingen wordt nagemaakte testdata gebruikt. In de integratie- en performanceomgevingen wordt gegenereerde testdata of eventueel geanonimiseerde productiedata gebruikt. Deze richtlijnen zijn bedoeld om privacy te waarborgen.

Er worden drie soorten data onderscheiden:

* **Statische data:** Dit is data die noodzakelijk is voor het correct functioneren van de applicatie. Denk hierbij aan systeemlocaties, certificaten, etc.;
* **Dynamische data:** Dit is data die voortkomt uit de interactie tussen gebruiker en systemen (applicaties);
* **Testgegevens:** Deze data wordt gebruikt om de test correct uit te kunnen voeren en wordt gecreëerd aan het begin van een testscript. Na het uitvoeren van de test wordt deze data weer verwijderd.

{Neem hier eventuele bijzonderheden op over testdata in het project en beschrijf wie verantwoordelijk is voor het maken en onderhouden van de testdata.}

# Rapportages en testartifacten

Dit hoofdstuk beschrijft de testrapportages en andere testartifacten die het project oplevert.

## Interne testrapportages

{Benoem hier de interne testrapportages die voor het project van toepassing zijn.}

De resultaten van de testuitvoer worden gedeeld met de software delivery manager in de vorm van de volgende interne testrapportages:

* **ART-resultaten:** Zichtbaar in Jenkins als resultaat van de Jenkins job.
* **Performancetestrapport:** Op wekelijkse basis, met hierin een overzicht van de testresultaten van de performancetest, de performancetrend, een analyse en eventueel advies.
* **Securitytestrapport:** De securitytesten worden uitgevoerd door een gespecialiseerde organisatie. Deze organisatie stelt ook het securityeindrapport op. Hierin staat een overzicht van de kwetsbaarheden die zijn ontdekt tijdens het testen, en een advies hoe deze te aan te pakken.

**Gebruikskwaliteitsrapport:** In dit rapport is vastgelegd in welke mate de software aan de eisen ten aanzien van gebruikskwaliteit voldoet.

* **Toegankelijkheidstestrapport:** In dit rapport is vastgelegd in welke mate de software aan de toegankelijkheidseisen voldoet, waaronder de wettelijk verplichte Web Content Accessibility Guidelines.

## Externe testrapportages

{Benoem hier de externe testrapportages die voor het project van toepassing zijn.}

De volgende externe testrapportages zijn onderdeel van elke release:

* **Eindrapport functionele testen:**
* Opsomming van de user stories per product,
* Gerelateerde logische testgevallen,
* Reviewstatus van de logische testgevallen,
* Of de testgevallen zijn geautomatiseerd,
* Wat het resultaat van de test was (geslaagd/gefaald/overgeslagen).
* **Performancetestrapport:** Een overzicht van de testresultaten van de performancetest, de performancetrend, een analyse en eventueel advies.
* **Securitytestrapport:** De securitytesten worden uitgevoerd door een gespecialiseerde organisatie. Deze organisatie stelt ook het securityeindrapport op. Hierin staat een overzicht van de kwetsbaarheden die zijn ontdekt tijdens het testen, en een advies hoe deze te aan te pakken.

**Gebruikskwaliteitsrapport:** In dit rapport is vastgelegd in welke mate de software aan de eisen ten aanzien van gebruikskwaliteit voldoet.

* **Toegankelijkheidstestrapport:** In dit rapport is vastgelegd in welke mate de software aan de toegankelijkheidseisen voldoet, waaronder de wettelijk verplichte Web Content Accessibility Guidelines.

## Bevindingenprocedure

{Beschrijf de bevindingenprocedure, zoals die voor het project geldt.}

## Testartifacten

{Benoem hier de testartifacten die voor het project van toepassing zijn.}

De volgende testartifacten worden bij een release opgeleverd:

* Broncode van de ART,
* Performance testrapportage,
* Security testrapportage,
* Gebruikskwaliteitsrapport,
* Toegankelijkheidstestrapport,
* Eind(test)rapport,
* Kwaliteitsrapport,
* Vrijgaveadvies.

Bijlagen

1. Terminologie en afkortingen

De onderstaande tabel bevat afkortingen en termen die voorkomen in de ICTU Kwaliteitsaanpak Softwareontwikkeling en bijbehorende templates.

|  |  |
| --- | --- |
| Term/afkorting | Toelichting |
| **actor** | een persoon die, of een extern informatiesysteem dat, een handeling verricht op het **informatiesysteem** |
| **architectuur** | een beschrijving van de structuur van een systeem, inclusief onderdelen, relaties tussen die onderdelen en eigenschappen van die onderdelen en relaties |
| **API** | application programming interface |
| **ART** | automatische **regressietest** |
| **auditing** | Vastlegging van de door een actor verrichte handelingen |
| **authenticatie** | het vaststellen van de identiteit van een **actor** |
| **autorisatie** | aan een **actor** toegekende rechten |
| **beheerorganisatie** | een (samenwerkingsverband van) organisatie(s) die in opdracht van een **opdrachtgevende organisatie** het **operationeel beheer**, applicatief beheer en/of functioneel beheer van **software** uitvoert |
| **BIA** | business impact analysis |
| **BIO** | Baseline Informatiebeveiliging Overheid |
| **broncode** | **software** in een vorm die leesbaar is voor mensen en de intentie van een programmeur uitdrukt |
| **deployment** | installatie van **software** op een systeem waardoor de software beschikbaar wordt gemaakt voor gebruik door **actor**en |
| **developers** | Developers zijn de mensen in het **Scrumteam** die iedere sprint gecommitteerd zijn aan het maken van elk aspect van een bruikbaar increment [Scrumgids] |
| **DevOps** | een praktijk die tot doel heeft **softwareontwikkeling** en **operationeel beheer** samen te brengen |
| **DoD** | definition of done |
| **DoR** | definition of ready |
| **gebruikskwaliteit** | mate waarin een systeem, product of dienst kan worden gebruikt door gespecificeerde gebruikers, voor het bereiken van gespecificeerde doelen, met effectiviteit, efficiëntie en tevredenheid in een gespecificeerde gebruikscontext |
| **GFO** | globaal functioneel ontwerp |
| **IB-plan** | informatiebeveiligingsplan |
| **informatiesysteem** | een samenhangend geheel van gegevensverzamelingen en de daarbij behorende personen, procedures, processen en **programmatuur** alsmede de voor het informatiesysteem getroffen voorzieningen voor opslag, verwerking en communicatie [VIR 2007, NORA] |
| **infrastructuurarchitectuur** | een **architectuur** die vooral de hardwareonderdelen en -relaties (housing, hardware, virtuals, standaard software en middleware) van een systeem beschrijft |
| **IPO** | intern projectoverleg |
| **ISD** | ICTU Software Diensten, afdeling van ICTU die **softwareontwikkelprojecten** ondersteunt met ontwikkel- en testomgevingen, tools en diensten |
| **ISE** | ICTU Software Expertise, afdeling van ICTU die **softwareontwikkelprojecten** ondersteunt met expertise op het gebied van **softwareontwikkeling** en die de ICTU Kwaliteitsaanpak Softwareontwikkeling onderhoudt |
| **ISO** | International Organization for Standardization |
| **Jira** | tool om **use cases**, user stories, logische testgevallen en issues vast te leggen |
| **klantreis** | alle directe en indirecte interactie van een klant of gebruiker met een product of dienst |
| **KPI** | key performance indicator |
| **kwaliteitsmanager** | controleert en borgt de kwaliteit van **software** conform de vastgestelde eisen en de Kwaliteitsaanpak en rapporteert aan de **projectleider** |
| **minimum viable product** | de eerste versie van een product of dienst, die zo vroeg mogelijk wordt uitgerold naar de gebruikers; het bevat net voldoende functionaliteit om het gestelde doel te behalen, en niet meer dan dat |
| **MTP** | master testplan |
| **MVP** | **minimum viable product** |
| **NFE** | niet-functionele eis(en) |
| **NORA** | Nederlandse Overheidsreferentie-architectuur |
| **NPR** | Nederlandse Praktijkrichtlijn |
| **ontwikkelaars** | Ontwikkelaars (*developers* in de Scrumgids) zijn de mensen in het **Scrumteam** die iedere sprint gecommitteerd zijn aan het maken van elk aspect van een bruikbaar increment [Scrumgids] |
| **opdrachtgevende organisatie** | overheidsorganisatie die opdracht geeft aan ICTU tot ontwikkeling en/of onderhoud van **software** |
| **opdrachtgever** | medewerker van de **opdrachtgevende organisatie** die eindverantwoordelijk is voor de opdracht aan ICTU |
| **operationeel beheer** | activiteiten die zorgen dat software operationeel is en blijft, zoals het oplossen van incidenten, het uitvoeren van onderhoud, het implementeren van upgrades en patches, het beheren van configuraties, en het monitoren van prestaties en beschikbaarheid |
| **OTAP** | ontwikkel, test, acceptatie, productie; gebruikt om verschillende soorten omgevingen aan te duiden |
| **persona** | een min of meer realistische beschrijving van een fictief persoon, veelal met naam, persoonskenmerken, drijfveren en behoeften, die een groep gebruikers representeert en gebruikt wordt om te redeneren over de gewenste functionele en niet-functionele eigenschappen van de **software** |
| **PIA** | privacy impact assessment |
| **PKI** | public key infrastructure |
| **PRA** | productrisicoanalyse |
| **Product owner** | De product owner is verantwoordelijk voor het maximaliseren van de waarde van het product, dat het resultaat is van het werk van het **Scrumteam** [Scrumgids] |
| **programmatuur** | zie **software** |
| **project** | een tijdelijke organisatie voor het realiseren van een resultaat - bij ICTU bestaat een **softwareontwikkelproject** uit medewerkers van ICTU, de **opdrachtgevende organisatie**, beheerorganisatie en eventueel andere partijen |
| **projectleider** | medewerker eindverantwoordelijk voor het projectresultaat - bij ICTU-softwareontwikkelprojecten is de projectleider een medewerker van ICTU |
| **PSA** | De projectstartarchitectuur is een concreet en doelgericht ICT-architectuurkader waarbinnen het **project** moet worden uitgevoerd |
| **PvE** | programma van eisen |
| **Quality-time** | een door ICTU ontwikkeld, open source, geautomatiseerd kwaliteitssysteem |
| **realisatiefase** | fase van een **softwareontwikkelproject** waarin de **software** daadwerkelijk wordt gebouwd en onderhouden, en bij een **DevOps** werkwijze ook operationeel wordt beheerd |
| **regressietest** | test die na een wijziging controleert of niet-gewijzigde delen van een systeem nog steeds correct functioneren |
| **release notes** | een overzicht van de wijzigingen in een **release** |
| **release** | een voor gebruik vrijgegeven versie van de **software** |
| **SAD** | software-architectuurdocument |
| **Scrum** | Scrum is een lichtgewicht raamwerk dat mensen, teams en organisaties helpt om waarde te creёren door middel van adaptieve oplossingen voor complexe problemen [Scrumgids] |
| **Scrummaster** | De Scrummaster is verantwoordelijk voor het opzetten van **Scrum**, zoals staat beschreven in de Scrumgids [Scrumgids] |
| **Scrumteam** | Een Scrumteam bestaat uit één **Scrummaster**, één **product owner** en **ontwikkelaars** (*developers* in de Scrumgids) [Scrumgids] |
| **softwarearchitectuur** | een **architectuur** die vooral de softwareonderdelen en -relaties (processen, modules, interfaces, datamodel) van een systeem beschrijft |
| **software delivery manager** | organiseert het ontwikkelen en opleveren van **software** conform de vastgestelde eisen en de Kwaliteitsaanpak en rapporteert aan de **projectleider** |
| **software** | software is de verzameling instructies die bepalen wat een computer uitvoert en is uiteindelijk wat de gebruiker ziet, ervaart en waarmee hij interacteert |
| **softwareontwikkeling** | een activiteit die nieuwe **software** maakt en/of bestaande software aanpast |
| **softwareontwikkelproject** | een **project** dat de oplevering van **software** als enige of voornaamste projectresultaat heeft |
| **solution architectuur** | beschrijving van de gewenste oplossing van een specifiek probleem, of het eindresultaat van een **project** [NORA] |
| **technische schuld** | eigenschappen van de **software** die de lange-termijninzetbaarheid en onderhoudbaarheid bedreigen |
| **TVA** | threat and vulnerability assessment |
| **usability** | gebruiksvriendelijkheid |
| **use case** | een afgebakende eenheid van interactie tussen een **actor** en het systeem |
| **UX** | user experience |
| **VIR** | Voorschrift Informatiebeveiliging Rijksdienst |
| **VIRBI** | Voorschrift Informatiebeveiliging Rijksdienst Bijzondere Informatie |
| **VM** | virtual machine, virtuele machine |
| **voorfase** | fase van een **softwareontwikkelproject**, voorafgaande aan de **realisatiefase**, waarin de uitgangspunten, risico's en randvoorwaarden voor de realisatiefase worden bepaald en waarin wordt gezorgd dat aan de randvoorwaarden wordt voldaan en dat voor zoveel mogelijk risico's maatregelen getroffen zijn |
| **vrijgaveadvies** | advies om een **release** vrij te geven voor ingebruikname, met een testverslag dat tenminste alle nog openstaande testbevindingen en geconstateerde beveiligingsbevindingen bevat |

1. Bronnen

De onderstaande tabel verwijst naar regelmatig gebruikte bronnen.

|  |  |
| --- | --- |
| Bron | Toelichting |
| [BIO](https://bio-overheid.nl/media/1572/bio-versie-104zv_def.pdf) | Baseline Informatiebeveiliging Overheid. |
| [ISO 9241-210:2019](https://www.iso.org/standard/77520.html) | Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. |
| [NCSC ICT-beveiligingsrichtlijnen voor webapplicaties](https://www.ncsc.nl/documenten/publicaties/2019/mei/01/ict-beveiligingsrichtlijnen-voor-webapplicaties) | De ICT-beveiligingsrichtlijnen voor webapplicaties geven een leidraad voor veiliger ontwikkelen, beheren en aanbieden van webapplicaties en bijbehorende infrastructuur. |
| [NEN-ISO/IEC 25010:2011](https://www.nen.nl/nen-iso-iec-25010-2011-en-157265) | Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. |
| [NEN-ISO/IEC 27001:2017](https://www.nen.nl/nen-en-iso-iec-27001-2017-a11-2020-nl-265545) | Informatietechnologie - Beveiligingstechnieken - Managementsystemen voor informatiebeveiliging - Eisen |
| [NEN-ISO/IEC 27002:2017](https://www.nen.nl/nen-en-iso-iec-27002-2017-nl-245390) | Informatietechnologie - Beveiligingstechnieken - Praktijkrichtlijn met beheersmaatregelen op het gebied van informatiebeveiliging |
| [NEN 7510:2017](https://www.nen.nl/nen-7510-1-2017-a1-2020-nl-267179) | Informatiebeveiliging in de zorg. |
| [NEN NPR 5325:2017](https://www.nen.nl/npr-5325-2017-nl-238298) | Praktijkrichtlijn voor het overdragen van software. |
| [NEN NPR 5326:2019](https://www.nen.nl/npr-5326-2019-nl-262885) | Praktijkrichtlijn voor risicobeheersing bij softwareontwikkeling. |
| [NORA](https://www.noraonline.nl) | Referentiearchitectuur voor de Nederlandse Overheid. |
| [OWASP Top-10](https://owasp.org/www-project-top-ten/) | De OWASP Top-10 is een op consensus gebaseerd overzicht van de meest kritische beveiligingsrisico's voor webapplicaties. |
| [Scrumgids](https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Dutch.pdf) | De Scrum Gids - De Definitieve Gids voor Scrum: De Regels van het Spel. |
| [VIR 2007](https://wetten.overheid.nl/BWBR0022141/2007-07-01) | Besluit Voorschrift Informatiebeveiliging Rijksdienst 2007. |
| [VIRBI 2013](https://wetten.overheid.nl/BWBR0033507/2013-06-01) | Besluit Voorschrift Informatiebeveiliging Rijksdienst Bijzondere Informatie 2013. |
| [Wbni 2018](https://wetten.overheid.nl/BWBR0041515/2020-07-15) | Wet Beveiliging Netwerk- en Informatiesystemen. Beschrijft de meldplicht en de zorgplicht die van toepassing zijn op organisaties die vitaal zijn én op digitale dienstverleners. |

1. De ICTU Kwaliteitsaanpak Softwareontwikkeling

De overheid is in hoge mate afhankelijk van informatiesystemen voor de uitvoering van haar taken. Veel van die informatiesystemen zijn dusdanig specifiek dat de benodigde software “op maat” gemaakt moet worden. De totstandkoming van op maat gemaakte software is meestal een complex proces, waarin vele belangen en behoeften worden afgewogen en afgezet tegen de mogelijkheden die technologie biedt. Eenmaal operationeel zal een informatiesysteem verantwoord onderhouden moeten worden; behoeften en technologie veranderen in de loop van de tijd.

Overheidsprojecten waarin software wordt ontwikkeld of onderhouden kampen nog vaak met vertraging, budgetoverschrijding of een eindresultaat met te lage kwaliteit. Zo concludeerde de commissie-Elias in haar [eindrapport](https://www.tweedekamer.nl/sites/default/files/field_uploads/33326-5-Eindrapport_tcm181-239826.pdf): "De Rijksoverheid heeft haar ICT (Informatie- en communicatietechnologie)-projecten niet onder controle". Eén van de fundamentele problemen is dat de risico's, die inherent zijn aan softwareontwikkeling, door organisaties nog onvoldoende worden herkend, erkend en gemitigeerd. Dit terwijl de risico's bij de ontwikkeling van software, binnen het ICT-domein, algemeen bekend zijn en er ook voor veel risico's passende maatregelen bestaan.

ICTU heeft jarenlange ervaring met het realiseren van software en past de opgedane ervaring toe bij de ontwikkeling van nieuwe software. Die ervaring is vastgelegd in een werkwijze, deze “ICTU Kwaliteitsaanpak Softwareontwikkeling”, die telkens wordt aangepast en aangevuld op basis van de praktijk.

ICTU is ervan overtuigd dat het bouwen van duurzame software, die goed aansluit bij de behoeften van gebruikers en andere belanghebbenden, bijdraagt aan betere informatiesystemen en een betere dienstverlening door de overheid. Dienstverlening die betrouwbaar moet zijn voor burgers, bedrijven en ambtenaren. Om samen met opdrachtgevende organisaties passende oplossingen te realiseren ontwikkelt ICTU daarom software volgens een agile proces. En om de duurzaamheid en betrouwbaarheid te bevorderen besteedt ICTU standaard aandacht aan beveiliging, privacy, performance, gebruikskwaliteit en toegankelijkheid. De Kwaliteitsaanpak dient daarvoor als leidraad, maar de aanpak voorziet ook in mogelijkheden om het project en het eindproduct aan te passen aan de specifieke situatie.

Om projecten, die software realiseren volgens de Kwaliteitsaanpak, efficiënt en effectief te ondersteunen, heeft ICTU twee gespecialiseerde afdelingen in het leven geroepen. Deze afdelingen staan projecten bij door middel van kennis, menskracht en technische hulpmiddelen. Zo profiteren projecten van schaalgrootte en hergebruik van inzichten.

Met behulp van de ICTU Kwaliteitsaanpak Softwareontwikkeling heeft ICTU samen met andere overheden inmiddels enige tientallen projecten succesvol uitgevoerd. ICTU wil deze aanpak graag aanvullen met de ervaringen en geleerde lessen van andere organisaties en deze overdraagbaar maken en breder uitdragen. Om die reden stelt ICTU deze Kwaliteitsaanpak aan iedereen beschikbaar via <https://www.ictu.nl/kwaliteitsaanpak> en heeft zij, samen met normalisatie-instituut NEN en partijen uit overheid en markt, een praktijkrichtlijn “Risicobeheersing bij ontwikkeling en onderhoud van maatwerksoftware” [NEN NPR 5326:2019] gepubliceerd, die mede is gebaseerd op de ICTU Kwaliteitsaanpak Softwareontwikkeling.

De ICTU Kwaliteitsaanpak Softwareontwikkeling heeft drie doelstellingen:

1. Opdrachtgevende organisaties helpen bekende risico's bij softwareontwikkeling, zoals technische schuld, vertraging en defecten, zo veel mogelijk te voorkomen.
2. ICTU helpen om software te ontwikkelen die de missie van ICTU, namelijk bijdragen aan een betere digitale overheid, ondersteunt.
3. De overheid als geheel helpen bij het zo goed mogelijk ontwikkelen van software.

De Kwaliteitsaanpak zelf is geformuleerd in de vorm van maatregelen die elke software-ontwikkelende organisatie kan treffen om risico's van softwareontwikkeling te mitigeren en de kans op succesvolle softwareontwikkelprojecten te vergroten. De maatregelen zijn gebaseerd op geleerde lessen uit de praktijk van ICTU.

De Kwaliteitsaanpak is een evoluerende aanpak, gebaseerd op de ervaringen die ICTU continu opdoet in de projecten waarin ICTU samen met opdrachtgevende organisaties maatwerksoftware ontwikkelt en onderhoudt. ICTU hanteert daarbij de vuistregel dat als tenminste 80% van de projecten minstens 80% van de tijd een bepaalde werkwijze hanteren, voor die werkwijze een maatregel in de Kwaliteitsaanpak wordt opgenomen. Maar het kan ook voorkomen dat maatregelen om andere redenen landen in de Kwaliteitsaanpak; denk aan het toegankelijk maken van software dat wettelijk verplicht is. Zie ook de wijzigingsgeschiedenis in [PDF-formaat](https://ictu.github.io/Kwaliteitsaanpak/v4.0.4/ICTU-Kwaliteitsaanpak-Wijzigingsgeschiedenis.pdf) of [HTML-formaat](https://ictu.github.io/Kwaliteitsaanpak/v4.0.4/ICTU-Kwaliteitsaanpak-Wijzigingsgeschiedenis.html).

De maatregelen vormen het startpunt voor de aanpak van ieder ICTU-softwareproject, waarbij ruimte wordt geboden voor variatie of alternatieve invulling. Bijvoorbeeld stelt de Kwaliteitsaanpak: software wordt minimaal bij iedere grote release of tenminste twee keer per jaar onderworpen aan een beveiligingstest door beveiligingsexperts die ICTU daarvoor inhuurt (zie M26: Het project laat de beveiliging van het ontwikkelde product periodiek beoordelen). Een alternatief is dat de opdrachtgevende organisatie de verantwoordelijkheid neemt voor het laten uitvoeren van beveiligingstests. Hierover maakt de projectleider nadere afspraken met de opdrachtgever.

De Kwaliteitsaanpak is dus zowel voorschrijvend als beschrijvend. Voorschrijvend omdat ICTU verwacht dat projecten die maatwerksoftware ontwikkelen en onderhouden de aanpak toepassen, en alleen aanpassen als daar een goede reden voor is, en mits dat wettelijk is toegestaan. Tegelijkertijd is de aanpak beschrijvend omdat de meeste maatregelen voortkomen uit de bestaande werkwijzen van de projecten. Zoals blijkt uit de self-assessment die ICTU regelmatig uitvoert op de toepassing van de Kwaliteitsaanpak.