

Compacte Voorfase

**{Productnaam, versie}**

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 wip, 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

{Beschrijf context en aanleiding voor dit project/product}

Dit document beschrijft de functionele en niet-functionele eisen voor {het product}. Daarnaast geeft het een eerste schets van datamodel en software architectuur van {het product} en de testaanpak. Het plan van aanpak, tenslotte, beschrijft de benodigde activiteiten en doet een inschatting van de benodigde tijd.

## Doelgroep

Dit document is in eerste instantie bestemd voor medewerkers van {opdrachtgevende organisatie} die {het product} laat ontwikkelen. Het biedt ook inzicht en overzicht aan andere betrokkenen bij het project, namelijk:

* de ontwikkelaar(s) van het systeem,
* de beheerder(s) van het systeem.

## Kaders

De volgende kaders zijn van toepassing op het softwareontwikkelingsproces.

|  |  |
| --- | --- |
| Volgnummer | Kader |
| K01 | NEN-ISO/IEC 27001:2017 en NEN-ISO/IEC 27002:2017 |
| K02 | VIR 2007, VIRBI 2013 en BIO voor het inrichten en beheren van informatiebeveiliging in brede zin. |
| K03 | {Indien van toepassing:} NEN 7510:2017 - Informatiebeveiliging in de zorg. |
| K04 | {Indien van toepassing:} Wbni 2018 - Wet Beveiliging Netwerk- en Informatiesystemen. |
| K05 | ICTU Kwaliteitsaanpak Softwareontwikkeling wip |

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 | {uitgangspunt} |
| U02 | {uitgangspunt} |
| U03 | {uitgangspunt} |
| {volgnummer} | {uitgangspunt} |

## Relatie met andere documenten

De volgende documenten waren input voor dit document:

* {Input documenten, bijvoorbeeld opdracht memo, Jira epic}
* {Context documenten, bijvoorbeeld bestaande documentatie}

## Leeswijzer

Hoofdstuk 2 beschrijft de functionele eisen aan {het product}. Hoofdstuk 3 beschrijft de niet-functionele eisen. Hoofdstuk 4 beschrijft de softwarearchitectuur. Hoofdstuk 5 beschrijft de testaanpak. Hoofdstuk 6 beschrijft het plan van aanpak.

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.

# Functionele eisen

Voor {het product} gelden de volgende functionele eisen, geformuleerd in de vorm van epics. Eisen voor het MVP zijn vetgedrukt.

## Hoofdfunctie 1 - {naam hoofdfunctie, bijvoorbeeld: Softwareproducten registreren}

{Introductie op hoofdfunctie 1, bijvoorbeeld: Beheerders kunnen softwareproducten registreren, inclusief de versies die beschikbaar zijn bij de leverancier.}

1. **{Titel epic 1, bijvoorbeeld: Registratie van softwareproducten}**: Als {rol} wil ik {actie} zodat {doel}. {Bijvoorbeeld: Als beheerder wil ik registreren welke softwareproducten ISD beheert zodat ik installaties van het softwareproduct kan registreren.}
2. **{Titel epic 2, bijvoorbeeld: Link naar beschikbare versies vastleggen}**: Als {rol} wil ik {actie} zodat {doel}. {Bijvoorbeeld: Als beheerder wil ik vastleggen via welke URL ik kan vinden wat de beschikbare versies van een softwareproduct zijn zodat ik kan registreren welke versies van het softwareproduct beschikbaar zijn.}
3. **{Titel epic 3, bijvoorbeeld: Beschikbare versies registreren}**: Als {rol} wil ik {actie} zodat {doel}. {Bijvoorbeeld: Als beheerder wil ik registreren welke versies van welke softwareproducten bij de leverancier beschikbaar zijn zodat ik kan zien welke software-installaties bijgewerkt kunnen worden.}
4. {Titel epic 4, bijvoorbeeld: Notificatie over nieuwe versies}: {Bijvoorbeeld: Als beheerder wil ik een notificatie ontvangen als er een nieuwe beschikbare versie van een softwareproduct geregistreerd wordt door een collega zodat ik niet zelf regelmatig hoef te kijken of er nieuwe versies beschikbaar zijn. }

## Hoofdfunctie 2 - {naam hoofdfunctie, bijvoorbeeld: Softwareinstallaties registreren}

{Introductie op hoofdfunctie 2, bijvoorbeeld: Bij installatie van een softwareproduct registreren beheerders welke versie van het softwareproduct op welke apparaat wordt geïnstalleerd. Dat maakt het mogelijk te zien wat waar is geïnstalleerd en de geïnstalleerde versies te vergelijken met de beschikbare versies.}

1. **{Titel epic 1, bijvoorbeeld: Registratie van software-installaties}**: Als {rol} wil ik {actie} zodat {doel}. {Bijvoorbeeld: Als beheerder wil ik registreren dat een versie van een softwareproduct op een apparaat is geïnstalleerd zodat ik kan zien welke versies van welke softwareproducten op welke apparaten zijn geïnstalleerd.}
2. **{Titel epic 2, bijvoorbeeld: Overzicht van software-installaties}**: Als {rol} wil ik {actie} zodat {doel}. {Bijvoorbeeld: Als beheerder wil ik bekijken welke versies van welke softwareproducten op welke apparaten zijn geïnstalleerd zodat ik in het geval van nieuwe bekende beveiligingskwetsbaarheden in bepaalde versies van een softwareproduct kan zien op welke apparaten deze versies van het softwareproduct zijn geïnstalleerd.}
3. **{Titel epic 3, bijvoorbeeld: Geïnstalleerde versies vergelijken met beschikbare versies}**: Als {rol} wil ik {actie} zodat {doel}. {Bijvoorbeeld: Als beheerder wil ik de versies van software-installaties vergelijken met de beschikbare versies van de softwareproducten zodat ik kan bepalen of ik een software-installatie kan bijwerken.}

## Hoofdfunctie 3 - {naam hoofdfunctie}

{Vul aan}

# Niet-functionele eisen

Onderstaande niet-functionele eisen zijn van toepassing op {het product}:

1. De oplossing is in het Nederlands omdat {rationale}.
2. De oplossing is compatible met {andere software}.
3. De oplossing wordt open source ter beschikking gesteld via {bijvoorbeeld: GitHub en PyPI (Python Package Index)}.
4. De oplossing voldoet aan de A- en AA-eisen uit WCAG 2.2.
5. De oplossing ondersteunt {...}.

{Vul aan.}

# Softwarearchitectuur

## Applicatiecontext

{Beschrijf in deze paragraaf de technische context waarbinnen de applicatie zal opereren. Denk aan omliggende systemen, authenticatie, logging en backup. Als deze informatie al in een ander document beschreven staat kan daarnaar verwezen worden uiteraard.}

## Datamodel

Het datamodel voor {het product} moet de hieronder beschreven entiteiten ondersteunen.

### Entiteit 1 - {naam entiteit, bijvoorbeeld: Softwareproduct}

{Omschrijving van de entiteit, attributen en relaties. Bijvoorbeeld: Een softwareproduct is een applicatie, firmware of besturingssysteem dat geïnstalleerd kan worden op een apparaat. Softwareproducten hebben een naam, een omschrijving en een lijst van tags. Softwareproducten hebben (net als device types) een manufacturer. Softwareproducten hebben een link naar een pagina op een website van de manufacturer met beschikbare versienummers. Softwareproducten hebben nul of meer versies.}

### Entiteit 2 - {naam entitieit, bijvoorbeeld: Softwareinstallatie}

{Omschrijving van de entiteit, attributen en relaties. Bijvoorbeeld: Een software-installatie is een softwareproduct dat geïnstalleerd is op één specifiek apparaat. Software-installaties hebben een versie, een installatiedatum, een commentaarveld en een lijst van tags. In NetBox kunnen devices services bevatten (services zijn combinaties van protocol en poortnummers en optioneel IP-adressen). Voor softwareproducten ondersteunen we geen services. Gebruikers kunnen bij installatie alleen versienummers invoeren die bij het softwareproduct voorkomen als beschikbaar versienummer.}

### Entiteit 3 - {naam entiteit}

{Vul aan}

## Componenten

{Beschrijf de belangrijkste componenten van het product. Denk aan database, webserver/services, models en views.}

## Koppelingen

{Beschrijf met welke systemen het product communiceert, welke informatie wordt uitgewisseld en welke protocollen en standaarden daarbij van toepassing zijn.}

## Deployment

{Beschijf hoe en waar het product zal worden gedeployed. Besteed eventueel ook aandacht aan migratie.}

## Implementatie

{Beschrijf programmeertaal en technologiestack voor de implementatie.}

# Testaanpak

{Beschrijf de testaanpak in termen van testsoorten, testtools, testomgevingen en betrokkenen.}

# Plan van aanpak

{Beschrijf beoogde resultaten, scope, werkwijze, planning, doorlooptijd en samenstelling van het team.}

De volgende activiteiten zijn nodig voor het ontwikkelen van de MVP van {het product}:

|  |  |  |
| --- | --- | --- |
| Nr. | Activiteit | Dagen |
| A01 | {Activiteit 1, bijvoorbeeld: Voorbereiding} | {X} |
| A02 | {Activiteit 2, bijvoorbeeld: Realisatie} | {X} |
| A03 | {Activiteit 3, bijvoorbeeld: Toegankelijkheidstoets} | {X} |
| A{X} | {Activiteit X} | {X} |
|  | Totaal | {X} |

Uitgangspunt is dat alle activiteiten door {X} ontwikkelaar(s) worden uitgevoerd en dat {activiteiten X en Y, bijvoorbeeld de daadwerkelijke uitrol naar de productieomgeving} buiten scope van dit plan van aanpak zijn.

Er zijn {X} eisen geïdentificeerd die buiten de scope van het MVP vallen (eisen {Y en Z}). De schatting nu is dat het realiseren van deze eisen {X} dagen extra zal kosten, maar uiteraard is een veel betere schatting mogelijk na realisatie van het MVP.

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** | Een business impact analyse is een methode om de mogelijke bedrijfsimpact te bepalen die een organisatie zou kunnen ervaren door een incident, dat de functionaliteit van of de informatie in een applicatie in gevaar brengt [NORA] |
| **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** | Een globaal functioneel ontwerp beschrijft de functionele werking van een product op hoofdlijnen, voor specifieke use cases |
| **IB-plan** | Een informatiebeveiligingsplan beschrijft binnen welke kaders bescherming geleverd wordt tegen welke dreigingen en met welke maatregelen die bescherming vorm krijgt |
| **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** | De infrastructuurarchitectuur beschrijft de technische infrastructuur van een product op hoofdlijnen, in termen van hardwareonderdelen en -relaties (housing, hardware, virtuals, standaard software en middleware) |
| **interactie-ontwerp** | Een interactie-ontwerp beschrijft de interacties tussen gebruikers en het systeem en de user experience daarbij |
| **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** | Een minimum viable product is een eerste versie van een product, die zo vroeg mogelijk wordt uitgerold naar de gebruikers, met net voldoende functionaliteit om het gestelde doel te behalen en niet meer dan dat |
| **MTP** | Een mastertestplan beschrijft de aanpak van het testen van een product op hoofdlijnen, in termen van strategie, activiteiten, afhankelijkheden en de op te leveren resultaten |
| **MVP** | **minimum viable product** |
| **NFE** | Niet-functionele eisen specificeren criteria om de kwaliteit van de software te beoordelen |
| **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** | Een privacy impact assessment geeft bij een wet of project, waar persoonsgegevens van toepassing zijn, aan wat de gevolgen voor de privacy van de getroffen personen zijn [NORA] |
| **PKI** | public key infrastructure |
| **PRA** | Een productrisicoanalyse is een analyse van het te testen product die resulteert in een overzicht van wat de meer of minder risicovolle kenmerken en delen van het te testen product zijn, zodat de grondigheid van testen hieraan gerelateerd kan worden |
| **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 [NORA] |
| **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** | Een software-architectuurdocument beschrijft de technische werking van een product op hoofdlijnen, in termen van softwarecomponenten, hun functies en hun onderlinge interacties en samenhang voor specifieke use cases |
| **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** | Een threat and vulnerability assessment inventariseert de betrouwbaarheidseisen die aan de bedrijfsprocessen en dientengevolge aan het product worden gesteld, gevolgd door identificatie en analyse van bedreigingen |
| **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/wip/ICTU-Kwaliteitsaanpak-Wijzigingsgeschiedenis.pdf) of [HTML-formaat](https://ictu.github.io/Kwaliteitsaanpak/wip/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.