

Software-architectuurdocument

**{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 eventueel deze paragraaf en de rubricering op de titelpagina indien 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} |

###### Revisiehistorie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versie | Auteur | Datum | Status | Opmerkingen |
| {versie} | {naam} | {datum} | {concept/definitief} | {opmerkingen} |

###### Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Functie/rol | Naam | Datum | Versie |
| Kwaliteitsmanager {opdrachtgever} | {naam} | {datum} | {versie} |
| Kwaliteitsmanager {beheerorganisatie} | {naam} | {datum} | {versie} |
| Kwaliteitsmanager ICTU | {naam} | {datum} | {versie} |

###### Vereiste goedkeuringen

|  |  |  |  |
| --- | --- | --- | --- |
| Functie/rol | Naam | Datum | Versie |
| Projectleider {opdrachtgever} | {naam} | {datum} | {versie} |
| Projectleider {beheerorganisatie} | {naam} | {datum} | {versie} |
| Projectleider ICTU | {naam} | {datum} | {versie} |
| Product owner | {naam} | {datum} | {versie} |

###### Verzendlijst huidige versie

|  |  |  |
| --- | --- | --- |
| Naam | Organisatie | Functie/rol |
| {naam} | {opdrachtgever} | Projectleider |
| {naam} | {opdrachtgever} | Product owner |
| {naam} | {beheerorganisatie} | Projectleider |
| {naam} | ICTU | Projectleider |
| {naam} | ICTU | Software delivery manager |

###### Template versie

Versie 4.0.0-dev, 27-11-2023

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

# Managementsamenvatting

{Managementsamenvatting}

# Inleiding

## Over dit document

Dit Software-architectuurdocument (SAD) beschrijft de software-architectuur van {de applicatie}. Het beschrijft op hoofdlijnen de softwarecomponenten, hun functies en hun onderlinge interacties en samenhang voor specifieke use cases. Het SAD heeft niet het detailniveau van een technisch ontwerp; dat wordt tijdens de realisatie verder uitgewerkt.

## Doelgroep

Dit document is bedoeld voor iedereen die kennis wil of behoort te hebben van de software-architectuur van {de applicatie}, waaronder IT-architecten, ontwikkelaars en testers.

## 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 | WCAG2.1 (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

Dit document beschrijft de software-architectuur. De architectuur die ten grondslag ligt aan de oplossing staat beschreven in de projectstartarchitectuur (PSA). De functionele beschrijving van de oplossing staat in het globaal functioneel ontwerp (GFO).

In verschillende documenten zijn eisen en wensen opgenomen die als basis dienen voor de software-architectuur. Deze zijn opgenomen in de volgende documenten:

* informatiebeveiligingsplan (bevat eisen en wensen en mogelijk maatregelen tegen informatiebeveiligingsrisico’s) {documentreferentie},
* projectstartarchitectuur (PSA) {documentreferentie},
* globaal functioneel ontwerp (GFO), {documentreferentie},
* niet-functionele eisen (NFE) (deze kunnen leiden tot functionele user stories), {documentreferentie}.

## Leeswijzer

{Wat staat waar in dit document?}

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.

# Ontwerpbeslissingen

Dit hoofdstuk bevat de ontwerpbeslissingen die betrekking hebben op de software. Zij zijn de basis voor de in dit document uitgewerkte software-architectuur.

{Dit hoofdstuk bevat een aantal veelgebruikte ontwerpbeslissingen. Waar nodig moeten ze worden aangevuld of aangepast, waarbij de relaties met de door opdrachtgever en beheerorganisatie gedefinieerde eisen en architectuurkaders traceerbaar zijn.}

## Ontwerpprincipes

Hieronder volgt een aantal ontwerpprincipes die binnen software engineering als best practice gelden. Sommige principes, zoals "Abstractie", worden afgedwongen door het gebruik van de gangbare API’s en standaardcomponenten; andere principes spelen een rol bij het ontwerp en de implementatie van de applicatie en worden expliciet getoetst door de SIG/TÜViT-richtlijnen voor onderhoudbaarheid van de software en continu geëvalueerd door Quality-time, het kwaliteitssysteem van ICTU.

* **Eenvoud**: Software die simpel en begrijpelijk is, is beter onderhoudbaar, overdraagbaar en uitbreidbaar. Dit is een principe waaruit andere voortvloeien.
* **Modulariteit**: Software is op alle detailniveaus opgesplitst in zelfstandige modules (componenten), zodat de modules afzonderlijk aangepast en getest kunnen worden.
* **Abstractie**: Modules (componenten) zijn voorzien van een abstracte interface die beschrijft wat de module doet en hoe ermee gecommuniceerd kan worden, zodat de modules kunnen samenwerken zonder dat er een rechtstreekse afhankelijkheid ontstaat. Met andere woorden: interface en implementatie zijn van elkaar gescheiden en onderdelen zijn alleen afhankelijk van de interfaces.
* **Low coupling, high cohesion**: Modules zijn zo onafhankelijk van elkaar als mogelijk (low coupling); de interne onderdelen van een module hebben juist een grote samenhang (high cohesion). Low coupling zorgt ervoor dat modules individueel gewijzigd kunnen worden zonder negatieve effecten op de rest van de applicatie, vergelijkbaar met Abstractie. High coupling verhoogt de eenvoud en de onderhoudbaarheid van de applicatie.
* **Separation of concerns**: Elke module heeft precies één taak of verantwoordelijkheid ("concern").

De term "module" hierboven is generiek bedoeld en duidt op alle vormen waarin delen van de software kunnen worden gebundeld, zoals componenten, klassen en methoden.

## Gelaagde applicatiearchitectuur

De applicatiearchitectuur van het systeem is gebaseerd op de onderstaande lagen. Elke component maakt onderdeel uit van precies één laag.

1. Presentatielaag: deze laag is verantwoordelijk voor de presentatie van informatie aan de eindgebruiker. {Eventueel opsomming van componenten uit deze laag.}
2. Applicatie-serviceslaag: deze laag bevat de businesslogica. {Eventueel opsomming van componenten uit deze laag.}
3. Technische-serviceslaag: deze laag bevat de ondersteunende technische services. {Eventueel opsomming van componenten uit deze laag.}
4. Datalaag of gegevenslaag: deze laag is verantwoordelijk voor de opslag van gegevens. {Eventueel opsomming van componenten uit deze laag.}

Componenten uit een laag mogen alleen componenten uit dezelfde of een diepere laag (een laag met een hoger nummer in de opsomming) gebruiken. Het is niet noodzakelijk dat dit de direct onderliggende laag is, als er redenen zijn om een laag "over te slaan".

## Gebruik van bewezen componenten en standaarden

{De applicatie} is gebaseerd op industriestandaarden {opsomming van standaarden}. Om deze standaarden te implementeren gebruikt {de applicatie} gangbare componenten en bibliotheken, bij voorkeur die onderdeel uitmaken van beheerde platformen. Gebruik van gangbare, bewezen en actief onderhouden componenten verlaagt de beheerlast op langere termijn; gebruik van beheerde en volwassen componenten verlaagt beveiligingsrisico's.

De toegankelijkheid van de applicatie voldoet aan de WCAG 2.1 (Web Content Accessibility Guidelines).

## Gebruik van gangbare security-ontwerpprincipes

{De Applicatie} maakt gebruik van de volgende gangbare security-ontwerpprincipes:

* **Least common mechanism**: Minimaliseer gedeelde resources tussen verschillende functies of gebruikers.
* **Least privilege**: Ken de minimale rechten toe die nodig zijn voor de adequate uitvoering van elke rol en taak.
* **Complete mediation**: Controleer de rechten voor elke toegang tot een object.

## Gebruiker beslist

In situaties waarin het niet op voorhand duidelijk is wat de correcte reactie van de applicatie is, wordt de keus aan de gebruiker gelaten. Een voorbeeld is het langdurig uitblijven van een reactie bij opvragen van gegevens: bij het bereiken van de ingestelde maximale wachttijd, krijgt de gebruiker de mogelijkheid om de betreffende actie nogmaals uit te voeren.

# Applicatiecontext

{Beschrijf in dit hoofdstuk de technische context waarbinnen de applicatie zal opereren. Indien deze informatie al in de PSA of ander document beschreven staan kan dit hoofdstuk weggelaten worden.}

## Omliggende systemen

### Management & control

### Softwareprovisioning

### Identity- en authorization-provisioning

### Dataprovisioning

### Logging & auditing

### Backup & recovery

## Gebruikerssystemen

## Externe systemen

# Applicatiearchitectuur

## Componenten

{Figuur met alle componenten}

### Webserver

### Directoryserver

### Functieservices

## Logische View - Dynamisch

### Use cases

{Beschrijf hier alleen de voor de architectuur significante use cases. Verwijs voor de uitwerking van deze use cases en de overige use cases naar het GFO. Gebruik hier dezelfde ids en namen voor de use cases als in het GFO.}

# Informatiearchitectuur

## Uitwisseling van gegevens met externe systemen

### Koppelvlak {identiteiten- en autorisatiessysteem}

### Koppelvlak {naam 1}

### Koppelvlak {naam 2}

## Conceptueel gegevensmodel

## Conceptueel gegevensmodel metadata

## Logginginformatie

### Technische (applicatie)log

### Berichtenlog

### Audit log

### Performance log

# Deployment view

# Implementatieview

## Scope

Dit hoofdstuk beschrijft de algemene richtlijnen voor de implementatie van {de applicatie} en de belangrijkste aspecten met betrekking tot de realisatie van de componenten.

## Implementatierichtlijnen

### Programmeertaal en programmeeromgeving

Criteria voor de keuze van een programmeertaal voor de implementatie zijn:

1. De programmeertaal moet toekomstvast en gangbaar zijn.
2. De programmeertaal en run-timeomgeving kennen een redelijke mate van onafhankelijkheid van het onderliggend besturingssysteem.
3. Het totale softwareplatform met gebruikte tooling moet eenvoudig over te zetten zijn naar een andere leverancier.

### Technische applicatielogging

### Gebruik van exceptions

### Gebruik van transacties

### Gebruik van threads en processen

### Toepassing van frameworks en libraries

### Coding rules

## Implementatietechnologie

### Besturingssysteem en middleware

### Applicatiecode

### Testcode

## Voortbrengingsproces

## Kwaliteitsstraat

## Omgevingen

## Componenten – Implementatieview

# Security view

## Inleiding

Dit hoofdstuk vat de technische beveiliging van {de applicatie} samen. De maatregelen zijn gegroepeerd naar systeemfunctie en voorzien van verwijzingen naar maatregelen in de risicoanalyse ({referentie}) en het informatiebeveiligingsplan ({referentie}).

## Gegevensbeveiliging tijdens transport

|  |  |  |
| --- | --- | --- |
| Nr | Technische maatregel | Relatie IB-plan |
| 1 | {maatregel} | {relatie} |

## Gegevensbeveiliging opgeslagen gegevens

|  |  |  |
| --- | --- | --- |
| Nr | Technische maatregel | Relatie IB-plan |
| 1 | {maatregel} | {relatie} |

## Authenticatie en autorisatie

|  |  |  |
| --- | --- | --- |
| Nr | Technische maatregel | Relatie IB-plan |
| 1 | {maatregel} | {relatie} |

## Auditing en accounting

|  |  |  |
| --- | --- | --- |
| Nr | Technische maatregel | Relatie IB-plan |
| 1 | {maatregel} | {relatie} |

## Richtlijnen

Bij de ontwikkeling van de programmatuur worden de volgende richtlijnen in acht genomen:

* [OWASP Top-10](https://owasp.org/www-project-top-ten/),
* [NCSC ICT-beveiligingsrichtlijnen voor webapplicaties](https://www.ncsc.nl/documenten/publicaties/2019/mei/01/ict-beveiligingsrichtlijnen-voor-webapplicaties).
* [Baseline Informatiebeveiliging Overheid (BIO)](https://bio-overheid.nl/media/1572/bio-versie-104zv_def.pdf).
* [Software Development (SSD) van het Centrum Informatiebeveiliging en Privacybescherming](https://www.cip-overheid.nl/media/1101/grip-op-ssd-het-proces-v20.pdf)

## Overzicht van gebruikte certificaten

Certificaten (alle PKIoverheid):

1. {lijst van PKIoverheidcertificaten}

Certificaten van anderen:

1. {lijst van andere certificaten}

# Performance view

## Inleiding

Dit hoofdstuk vat de performancekarakteristieken van {de applicatie} samen en beschrijft hoe de benodigde throughput, response tijden en schaalbaarheid bereikt zullen worden.

## Belasting

{Beschrijf of verwijs naar de verwachte belasting van het systeem: aantallen gebruikers, aantallen transacties, hoeveelheden te bewaren data, etc.}

## Snelheid (time behavior)

{Beschrijf hoe de applicatie de gewenste response tijden en throughput gaat realiseren}

## Middelen beslag (resource utilization)

{Beschrijf welke en hoeveel middelen (denk aan storage, bandbreedte, rekencapaciteit) nodig zijn voor de vereiste performance}

## Schaalbaarheid

{Beschrijf hoe de applicatie horizontaal danwel verticaal schaalbaar is}

# {Overige architectureel-significante niet-functionele eigenschappen}

{Voeg indien nodig hoofdstukken toe voor andere architectuureel-significante niet-functionele eigenschappen van de applicatie}

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 verrichtte handelingen. |
| **authenticatie** | het vaststellen van de identiteit van een **actor** |
| **autorisatie** | aan een **actor** toegekende rechten |
| **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] |
| **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, opdrachtgever, 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 |
| **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 opdrachtgevers 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. Opdrachtgevers 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 opdrachtgevers 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.

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 opdrachtgever 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.