

Globaal Functioneel Ontwerp

**{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 {beheerpartij} | {naam} | {datum} | {versie} |
| Kwaliteitsmanager ICTU | {naam} | {datum} | {versie} |

###### Vereiste goedkeuringen

|  |  |  |  |
| --- | --- | --- | --- |
| Functie/rol | Naam | Datum | Versie |
| Projectleider {opdrachtgever} | {naam} | {datum} | {versie} |
| Projectleider {beheerpartij} | {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} | {beheerpartij} | Projectleider |
| {naam} | ICTU | Projectleider |
| {naam} | ICTU | Software delivery manager |

###### Template versie

Versie 3.0.1, 31-03-2023

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

# Managementsamenvatting

{Managementsamenvatting}

# Inleiding

## Over dit document

Dit globaal functioneel ontwerp (GFO) beschrijft op hoofdlijnen de functionele werking van {de applicatie}. Het GFO geeft inzicht in de manier waarop {gebruikers} {de applicatie} gebruiken en hoe {de applicatie} samenwerkt met andere applicaties in het applicatielandschap van {de opdrachtgever}. Het document bevat een globale beschrijving van wíe met de applicatie wát kan doen, in de vorm van use cases.

{Verwijder één van de volgende twee paragrafen:}

Bij elke release levert ICTU een versie van het GFO dat is aangepast aan wat daadwerkelijk is gebouwd. Daartoe geeft het document per use case weer of deze al is gerealiseerd. Ook bevat het GFO de tijdens de realisatiefase genomen of aangepaste ontwerpbesluiten.

Het GFO wordt tijdens de realisatiefase niet onderhouden. Bij overdracht van het onderhoud naar een andere partij zal ICTU in opdracht van de opdrachtgever en in overleg met de ontvangende partij een actueel GFO maken.

## Doelgroep

Dit document is bedoeld voor degenen die direct betrokken zijn bij de realisatie, het beheer en het onderhoud van {de applicatie}: product owners, analisten, ontwikkelaars en beheerders. Daarnaast is het GFO nuttig voor belanghebbenden bij de functionele werking van {de applicatie}, zoals domeinexperts, (vertegenwoordigers van) gebruikers en architecten, om een overzicht te krijgen van de functionaliteit van {de applicatie}.

## Kaders

De volgende kaders zijn van toepassing op de functionaliteit van {de applicatie}:

|  |  |
| --- | --- |
| Volgnummer | Kader |
| K01 | NORA |
| K02 | {vul aan} |

## Uitgangspunten

De volgende uitgangspunten zijn van toepassing op dit document:

|  |  |
| --- | --- |
| Volgnummer | Uitgangspunt |
| U01 | Voor het beschrijven van de functionele werking van {de applicatie} worden use cases gebruikt. |
| U02 | De use cases worden uitgewerkt tot op het niveau van het primaire scenario. Alleen essentiële alternatieve scenario's worden uitgewerkt. |
| {volgnummer} | {uitgangspunt} |

## Relatie met andere documenten

De eisen en wensen die ten grondslag liggen aan dit globaal functioneel ontwerp zijn beschreven als functionele eisen in de vorm van een product backlog met epics en user stories {documentreferentie of link naar Jira}, als niet-functionele eisen in de vorm van een NFE-document {documentreferentie}, en in het informatiebeveiligingsplan {documentreferentie}.

De architectuur, die ten grondslag ligt aan de oplossing, staat beschreven in de projectstartarchitectuur {documentreferentie}. De beschrijving van de architectuur van de oplossing staat in het software-architectuurdocument {documentreferentie}.

Het GFO zelf vormt input voor de testplannen {documentreferentie}.

## Leeswijzer

Hoofdstuk 3 beschrijft de context van de applicatie. In hoofdstuk 4 is de werking op hoofdlijnen beschreven. Hoofdstuk 5 bevat de use cases.

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.

# Context

Dit hoofdstuk beschrijft de context waarin {de applicatie} zal opereren aan de hand van de organisaties en mensen die {de applicatie} gebruiken, de processen die de applicatie ondersteunt en het applicatielandschap waarbinnen {de applicatie} functioneert.

## Betrokken organisaties en gebruikers

{Beschrijf kort de organisatie(s) en/of (groepen van) mensen die de applicatie gebruiken. Als de applicatie primair door andere systemen wordt gebruikt, beschrijf die dan ook. Verwijs eventueel naar andere documenten, zoals bijvoorbeeld de PSA, voor meer informatie.}

## Ondersteunde processen

{Beschrijf de (bedrijfs)processen die de applicatie ondersteunt. Overweeg daarvoor business- of proces use cases te gebruiken. Verwijs eventueel naar andere documenten, zoals bijvoorbeeld de PSA, voor meer informatie.}

## Applicatielandschap

{Beschrijf het applicatielandschap waarin de applicatie gaat functioneren. Geef daarbij een schematische voorstelling van de applicatie in het applicatielandschap, inclusief koppelingen met aanpalende systemen zoals basisregistraties. Verwijs eventueel naar andere documenten, zoals bijvoorbeeld het SAD, voor meer informatie.}

# Werking op hoofdlijnen

Dit hoofdstuk beschrijft de werking van {de applicatie} op hoofdlijnen. Het beschrijft de hoofdfuncties die het systeem biedt, de ondersteunende functies, de belangrijkste informatie die het systeem bevat, en de klantreizen.

## Hoofdfuncties

{Beschrijf de functionele werking van de applicatie op hoofdlijnen en geef per functie de belangrijke actoren weer. Maak onderscheid tussen menselijke actoren en systeemactoren. Gebruik de hier beschreven functies als basis voor de paragraafindeling in het volgende hoofdstuk.}

## Ondersteunende functies

{Beschrijf de belangrijke ondersteunende functies die in meerdere (hoofd)functies terugkomen. Denk aan functies voor authenticatie en autorisatie, gegevensvalidatie, notificaties en logging. Gebruik onder andere de niet-functionele eisen om te bepalen welke ondersteunende functies belangrijk zijn om te beschrijven.}

## Gegevensverzamelingen

{Beschrijf hier de belangrijkste gegevensverzamelingen in hoofdlijnen. Voeg een eventueel logisch gegevensmodel toe in een apart hoofdstuk of als bijlage.}

## Klantreizen

{Beschrijf aan de hand van persona's hoe de gebruikers (menselijke actoren) bij de applicatie uitkomen, hoe ze door het systeem reizen, en wat ze na interactie met het de applicatie doen. Overweeg een los hoofdstuk met de klantreizen als deze omvangrijk zijn. Laat deze paragraaf weg als de belangrijkste actoren andere systemen zijn.}

# Use cases

{Gebruik voor de beschrijving van use cases in de voorfase een methode als de “casual” variant van Cockburn ("Writing effective use cases", 2001). In de realisatiefase en/of voor overdracht naar een andere partij kan de “fully dressed” variant (of een andere methode voor uitgebreidere beschrijvingen) worden toegepast. Doe dit gezien de onderhoudslast alleen als de opdrachtgever en/of ontvangende partij daar expliciet om vraagt.}

{Leg bij een update in de realisatiefase vast welke use cases zijn gerealiseerd, eventueel met de rationale en met een verwijzing naar de ontwerpbesluiten (in het SAD) die daarvoor zijn genomen.}

Dit hoofdstuk beschrijft de use cases die {de applicatie} ondersteunt. Per use case worden de volgende aspecten benoemd:

* Id: Unieke identificatie van de use case.
* Naam: Korte aanduiding van de use case.
* Doel: Het resultaat dat de (belangrijkste) actor wil bereiken door middel van de use case.
* Actor(en): Een actor is een persoon of een systeem; de actor gebruikt {de applicatie} om zijn of haar doel te bereiken door middel van de use case.
* Precondities: Wat moet er waar zijn om de use case te kunnen starten?
* Trigger: Wat start de use case?
* Primaire scenario: Beschrijving van de interactie tussen actor en systeem waarmee het doel bereikt wordt.
* Opmerkingen: Eventuele toevoegingen en openstaande punten.

De use cases zijn gegroepeerd naar de functies zoals beschreven in het vorige hoofdstuk.

## {F1} - {naam functie 1}

Deze paragraaf bevat de use cases behorende bij functie "{functie}".

### {UC1.1} - {naam use case 1.1}

|  |  |
| --- | --- |
| Aspect | Beschrijving |
| Doel | {Wat wil de (belangrijkste) actor bereiken?} |
| Actor(en) | {Wie interacteren met het systeem?} |
| Precondities | {Wat moet er waar zijn om de use case te kunnen starten?} |
| Trigger | {Wat start de use case?} |
| Primaire scenario | {Wat zijn de stappen in het belangrijkste scenario van de use case} |
| Opmerkingen | {Eventuele toevoegingen en openstaande punten} |

### {UC1.2} - {naam use case 1.2}

{Kopieer voorgaande subparagraaf om meer use cases toe te voegen.}

## {F2} - {naam functie 2}

{Kopieer voorgaande paragraaf om meer functies toe te voegen.}

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** |
| **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 softwarebeheer 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 (bron: VIR 2007, NORA) |
| **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]. |
| **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** | projectstartarchitectuur |
| **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]. |
| **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, 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. |
| [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” (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.