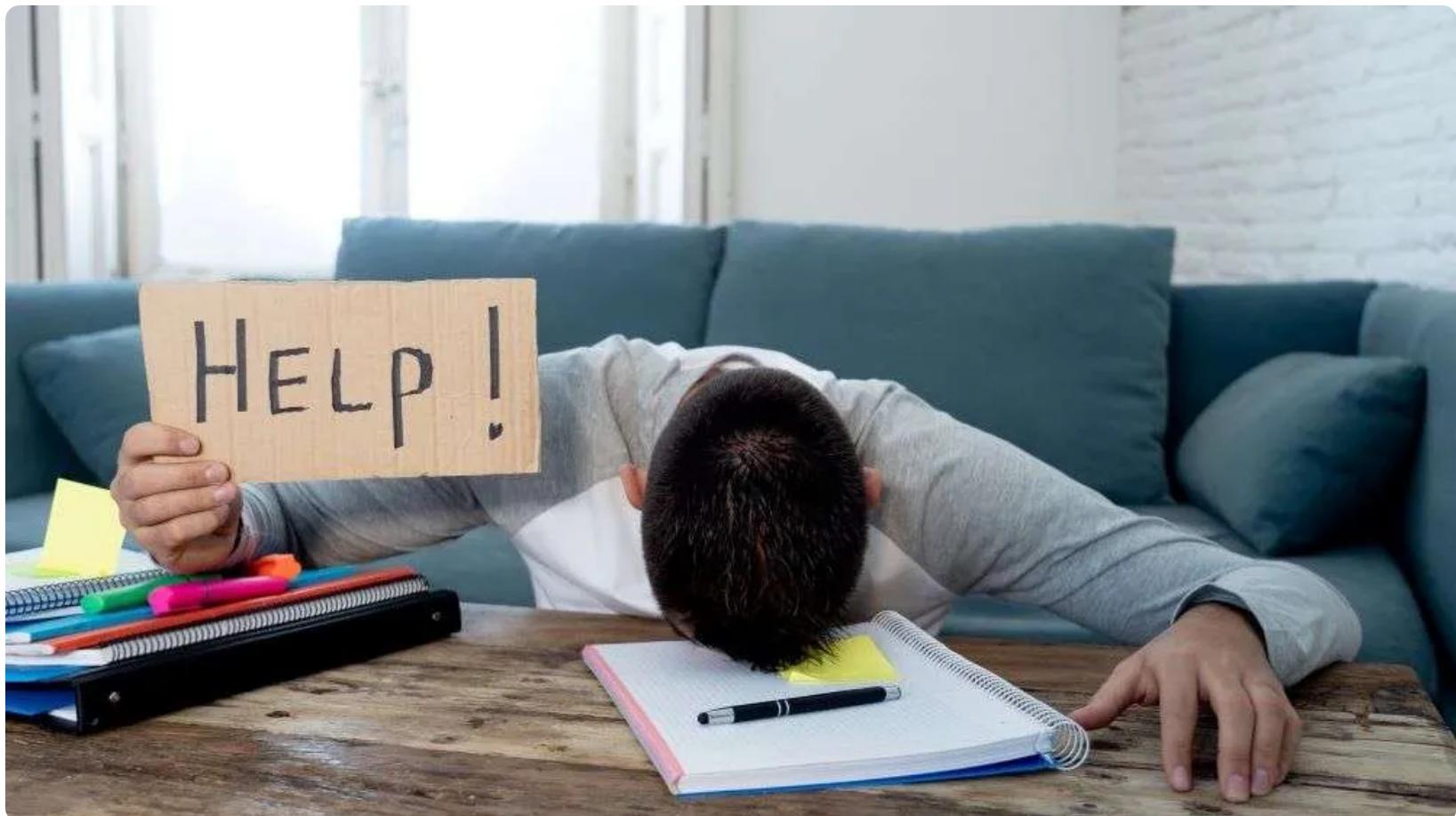


 [Terug](#)

Week 1: Van Overweldiging naar Vooruitgang



Ian Mondelaers

February 7, 2025

Share:



Het begin van mijn stage bij HolonCom markeert een significante transitie van de academische wereld naar professionele softwareontwikkeling. Deze week werd gekenmerkt door technische uitdagingen, leermomenten en waardevolle inzichten in enterprise-level ontwikkeling.

De Eerste Stappen

De aanpassing aan een nieuwe werkomgeving bracht initiële uitdagingen met zich mee. Het vinden van mijn weg in het kantoor, het leren kennen van collega's, en natuurlijk het lokaliseren van de koffieautomaat - essentieel voor elke developer - waren de eerste obstakels die overwonnen moesten worden.

Technische Aanpassingen en Uitdagingen

De overgang van Unix/Mac naar Windows-development was een significante verandering. Visual Studio verving mijn vertrouwde Rider IDE, en command-line Git maakte plaats voor een GUI-

interface. Het omschakelen van een QWERTY- naar AZERTY-toetsenbord voegde een extra laag complexiteit toe aan het typework. Deze aanpassingen, hoewel uitdagend, illustreren de flexibiliteit die van moderne developers verwacht wordt.

Het navigeren door een codebase met 135 C#-projecten was initieel overweldigend. Dit bood echter een unieke kans om te begrijpen hoe enterprise-level applicaties gestructureerd worden en hoe verschillende componenten met elkaar communiceren.

Code Kwaliteit en NuGet Expertise

De focus op code-kwaliteit werd direct duidelijk met de opdracht om null-reference warnings te elimineren. Dit werk benadrukte het belang van robuuste, veilige code in productieomgevingen.

Een belangrijk leerpunt was het werken met NuGet packages. Het proces omvatte:

- Het ontwikkelen van custom packages
- Lokale deployment en testing
- Integratie met bestaande projecten
- Het beheren van package dependencies
- Het schrijven van package-specifieke documentatie

Refactoring en Testing

Een significant project deze week betrof het refactoren van mail-gerelateerde functionaliteit. De uitdaging omvatte:

- Het verplaatsen van controller-logic naar een hoger abstractieniveau
- Implementatie van verbatim string-handling voor verbeterde code leesbaarheid
- Het schrijven van robuuste unit tests
- Het oplossen van cross-platform line-ending issues tussen Windows en Linux
- Het implementeren van helper functies voor consistente tekstverwerking

Automatisering en CI/CD

De laatste dagen focusten zich op het automatiseren van NuGet package deployment via Gitea. Dit proces omvatte:

- Het configureren van build actions
- Implementeren van semantic versioning conventies
- Het opzetten en configureren van lokale Gitea runners

- Debug symbol management voor verbeterde debugging mogelijkheden
- Automatische versie-incrementatie bij commits

Technische Groei en Reflectie

Deze eerste week heeft het verschil tussen academische en professionele softwareontwikkeling duidelijk gemaakt. De emphasis op code-kwaliteit, testing, en automation toont aan dat moderne softwareontwikkeling veel meer omvat dan feature development.

Het eerlijke gesprek met mijn mentor over de initiële overweldiging was waardevol. Het bevestigde dat het normaal is om je soms overweldigd te voelen door de complexiteit van enterprise software, en dat continue learning een essentieel onderdeel is van professionele groei.

Vooruitblik

Voor de komende weken ligt de focus op:

- Verdere automatisering van de NuGet deployment pipeline
- Uitbreiding van de mail-service functionaliteit
- Verdieping in enterprise-level testing strategieën
- Documentatie verbetering en standaardisatie

Comments

0 reactions



0 comments

© 2025 DenGian Blog. All rights reserved.

 [Terug](#)

[ProjectConfiguration](#)[DependencyManagement](#)[Refactoring](#)

Week 2: Bouwen aan Fundamenten



Ian Mondelaers

March 23, 2025

Share:



Na mijn eerste kennismaking met HolonCom, werd de tweede week gekenmerkt door diepere technische uitdagingen rond projectconfiguratie, buildprocessen en package management. Deze week bracht niet alleen technische groei, maar ook waardevolle inzichten in probleemoplossing en softwarearchitectuur.

Build-configuratie en Multi-Solution Management

De week begon met het repareren van de [Directory.Build.props](#) file die de vorige week problemen veroorzaakte. Een belangrijke les was dat naamconventies in .NET essentieel zijn - zelfs hoofdlettergebruik kan impact hebben op hoe het systeem bestanden vindt en implementeert.

Het werken met meerdere solutions naast elkaar was een interessante uitdaging. Ik leerde dat je via parameters en flags kunt bepalen welke solution gebruikt wordt tijdens het buildproces. Dit werd cruciaal bij het omzeilen van het DotNetNuke project dat buildproblemen veroorzaakte, wat we

oplosten door een tweede SLN file te maken zonder het problematische webproject.

ILRepack Verkenning en Hindernissen

Een aanzienlijk deel van de week besteedde ik aan het onderzoeken van ILRepack - een tool om meerdere DLL's samen te voegen tot één enkele DLL, die vervolgens als NuGet package gedistribueerd kan worden. De uitdaging hierbij was het gebrek aan documentatie en community support.

```
// Lokaal werkte de command line aanpak zoals:  
ilrepack /out:CombinedLibrary.dll /lib:path\to\dependencies Library1.dll Library2.dll Library3.dll
```

De echte uitdaging lag in het integreren van dit proces in Gitea actions. De cross-platform compatibiliteit werd een obstakel - ILRepack is primair ontworpen voor Windows, terwijl de Gitea runners op Ubuntu draaiden. Mijn poging om MONO te implementeren als oplossing kwam dicht bij succes, maar uiteindelijk besloten we na grondig onderzoek om deze aanpak te verlaten en andere oplossingen te verkennen.

Project Configuratie en Props Files Management

Een cruciale leerfase was het begrijpen van .NET props files en hoe deze elkaar beïnvloeden op verschillende niveaus. Ik ontdekte dat properties op projectniveau die op solutionniveau overschrijven, wat een flexibel maar soms complex systeem creëert.

De implementatie van dynamische referentiewisseling tussen online NuGet packages en lokale projectreferenties was een significant hoogtepunt:

```
<!-- Voorbeeld van conditionele referenties in project file -->
<ItemGroup Condition="'$(ReferenceType)' == 'nuget'">
    <PackageReference Include="HolonCom.Core" Version="$(PackageVersion)" />
</ItemGroup>
<ItemGroup Condition="'$(ReferenceType)' == 'source'">
    <ProjectReference Include="..\..\Core\HolonCom.Core\HolonCom.Core.csproj" />
</ItemGroup>
```

Deze aanpak werd later gerefactored om variabelen met descriptieve waarden te gebruiken in plaats van directe condities, wat de leesbaarheid en onderhoudbaarheid ten goede kwam.

Refactoring en Folderstructuur

Een groot deel van de week werd besteed aan het herstructureren van projectfolders en het

centraliseren van logica in de Directory.Build.props files. Dit vereiste nauwkeurige aanpassingen van paden en referenties tussen projecten.

De reorganisatie van HolonCom.Core en HolonCom.Abstractions in één repository was succesvol, maar niet zonder uitdagingen. Het aanpassen van actions (YAML) files, project files en README.md implementaties leerde me het belang van consistentie en best practices in package publicatie.

Caching Uitdagingen

Een terugkerend thema was de impact van caching op het ontwikkelproces. Zelfs na het aanpassen van foute paden konden builds soms slagen door gecachte data, wat later tot verwarringe fouten leidde. Deze ervaring onderstreepte een cruciale les in softwareontwikkeling:

""Caching issues blijven een van de grote problemen binnen IT. Het is essentieel om hier waakzaam op te zijn."""

Het systematisch leegmaken van de obj folder en soms zelfs de .VS folder bleek noodzakelijk, wat het belang van schone buildprocessen benadrukte.

Professionele Ontwikkeling

Naast technische groei bracht deze week ook persoonlijke ontwikkeling. Het gesprek met mijn stagmentor over race conditions in asynchrone programmeertalen evolueerde naar een diepgaand gesprek over persoonlijke groei en kritisch denken. Het concept van de "chain of trust" versus eigen ervaringen was bijzonder inzichtgevend en benadrukte het belang van kritisch denken in softwareontwikkeling.

De kennismaking met mijn AP Hogeschool-begeleider bracht duidelijkheid over de stage-evaluatieprocessen en belangrijke mijlpalen voor de komende maanden.

Vooruitblik

De komende weken zal ik me richten op het systematisch migreren van managed libraries uit de command worker naar afzonderlijke projecten die als NuGet packages worden gedistribueerd. Dit zal het gebruik van libraries flexibeler maken en het debugproces vereenvoudigen.

Comments

0 reactions



0 comments

Write Preview Aa

Sign in to comment

M▼

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.



DENGIAN

[Home](#)[Blog](#)[Over Mij](#)[Contact](#)[Log In](#) [Terug](#)

[NuGetDeployment](#)[CI/CD](#)[CodeObfuscation](#)

Week 3: Nuggets, Scripts en Debugging



Ian Mondelaers

March 23, 2025

Share:



Na twee weken bij HolonCom waarin ik de basis leerde kennen en werkte aan projectconfiguraties, werd mijn derde week gekenmerkt door diepere technische uitdagingen rondom NuGet packages, automatiseringsscripts en het debuggen van andermans code. Deze week bracht niet alleen technische groei, maar leerde me ook waardevolle lessen over geduld en systematisch probleemoplossen.

NuGet Packages en Projectrefactoring

De week begon waar ik geëindigd was: het project bouwde nog niet succesvol vanwege padproblemen. Bij de transitie van libraries naar absolute en relatieve paden binnen een gecentraliseerd project liep ik tegen diverse uitdagingen aan. In het core project implementeerde ik een schakelaar om flexibel te kunnen wisselen tussen NuGet packages en lokale projectreferenties - een elegante oplossing die onze ontwikkelwerkstroom aanzienlijk verbeterde.

Tijdens het troubleshooten van build-issues stuitte ik op de beruchte "ambiguous name" error. Na wat speurwerk ontdekte ik dat het probleem niet lag bij identieke namen van solution en project zoals ik eerst dacht, maar bij een dubbel gebruikte package name. Deze kleine maar cruciale aanpassing liet de build eindelijk slagen.

Het documenteren van versioning policies vormde een interessante uitdaging. Semantic versioning is essentieel voor betrouwbare softwareontwikkeling, maar het bleek dat onze stabiele versie (3.0.0) alleen in de testomgeving bestond. Dit inspireerde me om onze GitHub Actions workflow te verbeteren met nieuwe functionaliteiten:

- Automatisch triggeren van de action bij nieuwe tags in Gitea
- Bouwen van packages voor pull requests om te testen (zonder publicatie)
- Automatische Gitea releases bij het pushen van tags

Deze verbeteringen maakten onze deployment pipeline robuuster en beter geautomatiseerd, wat de efficiëntie van ons ontwikkelproces aanzienlijk verhoogde.

Code Obfuscation en Deployment Challenges

Een bijzonder leerzaam onderdeel van de week was het implementeren van code obfuscation met

"Obfuscar" - een techniek om reverse-engineering tegen te gaan door code te verwarren. Dit proces vereiste aanpassingen in de Global.props file en het toevoegen van XML-configuratiebestanden per project. Bij het testen ontdekte ik een interessant fenomeen: Windows CMD doet aan caching, waardoor ik voor elke test eerst de prompt moest afsluiten om accurate resultaten te krijgen.

Een cruciaal inzicht tijdens dit proces was het belang van de "CopyLocalLockFileAssemblies"-variabele, die op true gezet moest worden om dependencies correct te laten pakken. Deze kleine maar essentiële kennis illustreert hoe diepgaand het .NET ecosysteem kan zijn.

De workflow voor code obfuscation stuitte echter op een onverwachte hindernis: de Obfuscar.Console.exe is een Windows executable, maar onze Gitea runner draait op Ubuntu. Dit cross-platform probleem deed me denken aan de ILRepack uitdagingen uit week 2. Na overleg met mijn mentor besloten we om de obfuscated NuGet packages handmatig te deployen naar een nieuwe Gitea organisatie. Om dit proces te stroomlijnen, ontwikkelde ik met behulp van Cursor AI een script dat de deployment automatiseerde.

Netwerktroubleshooting en Remote Issues

Halverwege de week werd ik geconfronteerd met connectiviteitsproblemen na een Windows update. De Remote Desktop connectie naar onze VM faalde, wat resulteerde in een diepgaande troubleshooting sessie van netwerkconfiguratie, firewalls en RDP-protocol instellingen. Tijdens het wachten op een oplossing benutte ik de tijd door literatuur over gegevensleer te bestuderen - een

mooi voorbeeld van hoe downtime constructief kan worden gebruikt.

Uiteindelijk ontdekten we dat de automatische traffickingssnelheid in Remote Desktop race conditions veroorzaakte - een waardevolle les in netwerktroubleshooting die aantoonde dat problemen soms op onverwachte plekken zitten. Dit herinnerde me aan wat mijn mentor eerder zei over race conditions in asynchrone programmeertalen.

Van BAT naar PowerShell: Scriptautomatisering

Een belangrijk deel van mijn week bestond uit het verfijnen van scripts voor NuGet deployment. Aanvankelijk begon ik met .BAT bestanden voor Windows, maar mijn mentor adviseerde om over te stappen op PowerShell voor geavanceerdere functionaliteit. De transitie tussen deze scripting languages was een leerzame ervaring die mijn toolset verbreedde.

Het script moest dynamisch zijn met parameters voor versie, projecten, solution paden en NuGet bronnen. De grootste uitdaging was het correct beheren van alpha-versies en het pushen van verschillende versies naar verschillende repositories - Holoncom voor alpha-versies en Desco voor stabiele versies.

De refactoring van het script van een hardcoded implementatie naar een parametrische aanpak maakte het herbruikbaar voor verschillende projecten en scenarios. Ik creëerde uiteindelijk twee versies: één met relatieve paden en één met absolute paden, wat de flexibiliteit voor verschillende

ontwikkelomgevingen verhoogde.

Debugging Andermans Code: Het Timotheus Project

De laatste dagen van de week waren gewijd aan het debuggen van het Timotheus project - een klassiek voorbeeld van het werken met erfcode. De berekeningen voor de enneagram- en stresstesten werkten niet correct; ongeacht de gegeven antwoorden kwamen dezelfde resultaten terug, en de hash van de vragen klopte niet.

Door systematische unit tests te schrijven (meer dan 500 regels aan tests) en debug informatie te verwerken, kon ik geleidelijk de problemen identificeren. Bij de stresstest ontdekte ik bijvoorbeeld dat sommige antwoorden voor 500% meetelden en andere voor 0% - een duidelijke oorzaak van de incorrecte resultaten.

Deze debugging-ervaring leerde me het belang van methodisch werken bij het analyseren van andermans code. Het monitoren van console en netwerkverkeer via developer tools was essentieel om JavaScript errors en API-interactieproblemen te identificeren. Het bevestigde ook mijn mentors wijsheid over het belang van documentatie: goede code beantwoordt de "hoe" vraag, maar documentatie verklaart het "waarom" - cruciaal bij het navigeren door onbekende codebasissen.

Sociale Dimensie: Team Building

Een warm moment in de week was een team lunch, een traditie bij HolonCom voor nieuwe medewerkers. Hoewel het met twee weken vertraging plaatsvond, was het een gezellige gelegenheid om mijn collega's beter te leren kennen buiten de technische context van ons werk. Deze sociale dimensie van software ontwikkeling is net zo belangrijk als de code zelf en droeg bij aan mijn gevoel van verbondenheid met het team.

Reflectie en Leerpunten

Ondanks alle inspanningen eindigde de week met enkele onopgeloste problemen in het Timotheus project - een JavaScript error bij het doorsturen van antwoorden en incorrecte scoringsresultaten. Het voelde onbevredigend om iets "kapot" achter te laten voor het weekend, maar het is ook een realistische weerspiegeling van softwareontwikkeling: soms moet je accepteren dat niet alles in één sprint opgelost kan worden.

Deze derde week heeft me geleerd dat geduld, systematisch troubleshooten en goede communicatie met je team essentiële vaardigheden zijn voor een software engineer. Elke bug is een leermoment, elke commit een kleine overwinning, en samen vormen ze de weg naar groei en vakmanschap in dit veld.

Voor de komende week sta ik voor de uitdaging om de Timotheus issues volledig op te lossen en verder te gaan met de systematische migratie van managed libraries naar afzonderlijke, goed gedocumenteerde NuGet packages. Ik kijk uit naar de volgende stappen in mijn ontwikkeling als

software engineer bij HolonCom.

Comments

0 reactions



0 comments

Write Preview Aa

Sign in to comment

M▼

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#)[Blog](#)[Over Mij](#)[Contact](#)[Log In](#) [Terug](#)

[Refactoring](#)[Debugging](#)[ErrorHandling](#)[Vue.js](#)

Week 4: Probleemoplossend werken en refactoring

**Ian Mondelaers**

May 11, 2025

Share:



Deze week werd gekenmerkt door diepgaand debuggen, uitgebreide refactoring en een aantal belangrijke leermomenten, zowel technisch als op persoonlijk vlak.

Timotheus project afronden

De week begon met het verder werken aan het Timotheus project. Na een korte stand van zaken met mijn mentor (waarbij ik ook meldde dat ik woensdag afwezig zou zijn voor een intervisiesessie op school en de jobbeurs), ben ik in de codebase gedoken.

De logs toonden dat niet alle vragen werden beantwoord, wat zichtbaar was dankzij de error handling die ik vorige week had geïmplementeerd. Mijn mentor vroeg me om deze error handling te verbeteren voor een meer gedetailleerde logging, wat me zou helpen bij het debuggen.

Front-end vs Back-end problematiek

Hoewel alle tests in de back-end succesvol waren, bleek er toch iets mis te gaan. Bij grondige analyse via de devtools ontdekte ik dat er veel '0'-resultaten werden doorgestuurd. Ik kon de front-end code inspecteren, waar ik enkele interessante bevindingen deed:

1. De front-end (gebouwd in Vue.js) had eigenlijk nooit correct gewerkt
2. Er was een workaround gemaakt die null/undefined waardes omzette naar 0
3. In de back-end werden alle 0-waardes toegewezen aan een matige score (50%)

Dit verklaarde waarom er inconsistenties waren in de resultaten. Door deze bevindingen kon ik gerichter te werk gaan:

Na het verbeteren van de logging en het zorgen dat de data die werd doorgestuurd aan de DTO-structuur voldeed, merkte ik dat de HTML click-handler camelCase gebruikte in plaats van PascalCase zoals in de back-end. Dit zorgde ervoor dat antwoorden niet correct werden doorgegeven. Na het consistent maken van de naamgeving werkten de front-end overdrachten correct.

Als laatste stap heb ik ook de gegenereerde link op basis van de slechtste score gerepareerd en dynamisch gemaakt. Ook heb ik een manier gevonden om alle mogelijke testuitkomsten te tonen met behulp van twee tests. Deze resultaten worden weggeschreven naar HTML-bestanden voor

eenvoudige online visualisatie.

CommandWorker refactoring

Het tweede grote project van deze week was het refactoren van de CommandWorker. De folder structuur was aangepast waardoor ik in de SLN file en project files de absolute paden moest aanpassen. De dag begon hierdoor met maar liefst 2100 errors!

Systematisch heb ik alle projecten die in de core zaten overgezet en gerefactored. Dit proces omvatte:

1. Het verplaatsen van projecten naar de HolonCom.Core structuur
2. Het omzetten van absolute paden naar relatieve paden
3. Het updaten van referenties binnen de projecten

De scope van dit werk werd gedurende de week uitgebreid. Eerst moest ik alleen de core components (managed libraries) overzetten en er NuGet packages van maken, maar later kwamen daar ook alle drivers en app-gerelateerde code bij.

NuGet packages maken en deployment

Op vrijdag heb ik bewezen dat relatieve paden ook kunnen werken naar projecten buiten je solution. Vervolgens kreeg ik de taak om NuGet packages te maken en workflows op te zetten voor automatische deployment.

Dit betekende:

1. Admin rechten verkrijgen op de repositories
2. GitHub Actions/workflows creëren voor automatische packing en pushing bij elke push naar develop
3. Werken met SourceTree in plaats van Git via terminal (wat een aanpassing vereiste)
4. Repositories van GitHub naar Gitea verplaatsen, inclusief mergeconflicten oplossen

Een van de repositories had wat problemen - ik had onvoldoende rechten en er leken fouten in de codebase te zitten. Dit is een aandachtspunt voor volgende week.

Persoonlijke ontwikkeling

Naast alle technische uitdagingen, was er ook aandacht voor persoonlijke groei. In een gesprek van 1,5 uur met mijn mentor hebben we de beoordelingscriteria van school besproken. Mijn mentor gaf me enkele belangrijke aandachtspunten:

- Meer assertief zijn in discussies en overleggen
- Me kwetsbaar durven opstellen en leren van fouten
- Minder voorspelbaar zijn in mijn dagelijkse interacties (zoals bij het afscheid nemen)
- Mijn definitie van succes breder maken

Deze feedback neem ik serieus en ik ga er bewust aan werken in de komende weken.

Intervisie en jobbeurs

Op woensdag had ik een intervisiesessie op school met zeven andere stagiairs. Het was interessant om te zien hoe breed het IT-veld is - niemand werkte met exact dezelfde technologieën. Ik merkte dat ik de enige was die echt als software engineer werkt binnen mijn groep.

In de discussie deelde ik mijn inzicht dat het oké is om te falen, en dat zelfs ervaren ontwikkelaars fouten in hun code schrijven. We zijn allemaal gewoon mensen die ons best doen.

Na dit moment van intervisie ben ik naar de jobbeurs in Antwerpen Expo gegaan. Ondanks de drukte en warmte was dit een leerzame ervaring. Ik heb enkele waardevolle connecties kunnen leggen en met verschillende bedrijven gesproken. Ook was het fijn om klasgenoten tegen te komen en te horen dat iedereen het goed doet op zijn stage.

Conclusie

Het was een intensieve week met veel technische uitdagingen, maar ook belangrijke leermomenten. De combinatie van diepgaand debuggen, refactoring en persoonlijke ontwikkeling maakt deze stage tot een waardevolle ervaring.

Volgende week ga ik verder met het oplossen van de resterende errors in de core projecten en het afmaken van de NuGet deployment workflows.

Comments

0 reactions



0 comments

Write

Preview

Aa

Sign in to comment



Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

 [Terug](#)

Week 5: NuGet deployment, unit testing en een vleugje Timotheus



Ian Mondelaers

May 11, 2025

Share:



Deze week stond in het teken van het verder uitwerken van het Core-project, het uitgebreid testen van libraries en onverwachte verbeteringen aan het Timotheus project. Ook maakte ik kennis met AI-ondersteunde ontwikkeltools die mijn workflow drastisch veranderden.

Core.All project: NuGet packages en deployment

De week begon met het fixen van het Core.All project. Het hoofddoel was om de build weer werkend te krijgen, de action op Gitea te herstellen en de beschikbare NuGet packages te deployen.

Debugging en refactoring

Een lastig te vinden error bleek te schuilen in een verstopt .props-bestand in één van de

subprojecten, dat een niet-bestand icon.png verwachtte tijdens de NuGet build:

```
<!-- Probleem in props-file -->
<PropertyGroup>
    <PackageIcon>icon.png</PackageIcon>
    <!-- Andere properties -->
</PropertyGroup>
```

Na deze ontdekking refactorde ik het project zodanig dat er meer NuGet packages beschikbaar zouden komen. De volgende uitdaging was een external library met versie-incompatibiliteiten. Hier bleek een downgrade van .NET 9.0 naar .NET 8.0 een deel van de oplossing, maar er bleven errors bestaan.

Na overleg met mijn mentor hebben we besloten om niet-essentiële projecten uit de external library te verwijderen. Dit resulteerde in één project dat succesvol kon builden, waarna we de action konden configureren om deze solution te gebruiken tijdens het build-proces.

Versioning issues

Het koppelen van de NuGet packages van de external libraries aan het Core-project was complexer dan verwacht:

1. Eerst waren er versioning issues die gestroomlijnd moesten worden
2. De referenties naar de external libraries bleven bestaan in de solution
3. Er waren nog steeds referenties naar obsolete projecten
4. De Gitea link in de action was niet meer correct
5. Een README.MD bestand had een incorrecte naam

Na het oplossen van al deze issues werkte alles zoals verwacht en waren alle NuGet packages succesvol gedeployed.

Command Worker refactoring

Dinsdag ging ik verder met de Command Worker, waar ik enkele errors moest wegwerken. Het pad naar het project in de solution klopte niet meer. Na het fixen hiervan en het uitvoeren van de nodige refactoring om ook hier NuGet packages van te maken, werkte de build zoals verwacht.

De absolute paden in de solution heb ik omgevormd naar relatieve paden middels enkele find-and-replace operaties. Hierna werkte mijn mentor aan een clean-up job op Gitea om oude NuGet packages te verwijderen, die ik vervolgens moest controleren en verbeteren.

Keep the most recent:

5 versions per package

Keep versions matching:

(^\d+\.\d+\.\d+\$) | (.*-rc\..*)

Remove versions matching:

.*(alpha|beta|develop|preview|dev).*

Door middel van regex expressies heb ik de clean-up job verbeterd zodat alleen de meest recente alpha-versie en alle stabiele releases behouden blijven.

Unit Testing

De volgende grote opdracht was het schrijven van unit tests voor elk project in Core.All. Mijn mentor wilde dat ik in de testprojecten dezelfde structuur hanteerde als in de libraries, wat het proces aanzienlijk complexer maakte.

Structuur en merge-issues

Woensdag begon met het doornemen van de planning met mijn mentor. Hij had een merge

uitgevoerd van mijn branch en de develop branch van de Command Worker. De eerste stap was het verwijderen van bin/obj folders en lege directories met een script, gevolgd door het rebuilden van beide solutions.

Er doken verschillende structuurproblemen op omdat sommige projecten niet meer in de source map zaten. Hierdoor kwamen testfolders onder de root van het project terecht en werd de project file zonder de juiste package references gebruikt. Na het oplossen van deze issues kon ik de stabiele versies online pushen.

Unit tests en code issues

Het schrijven van unit tests bracht enkele fouten in de libraries aan het licht. Na overleg met mijn mentor besloten we de libraries aan te passen en nieuwe stabiele versies te deployen. Een specifiek probleem betrof de ASCII-encoding, aangezien we nu Base64 gebruikten en er inconsistenties optraden bij karakters boven 127 bytes (ASCII is 7-bits), wat resulteerde in corrupte gegevens.

Cursor: AI in development

Woensdag begon ik ook met het gebruik van Cursor, een AI-tool die kan compileren, tests runnen, commando's uitvoeren en code schrijven. Het was een bijzondere ervaring die mijn productiviteit enorm verhoogde - ik heb binnen een dag ongeveer 400 tests kunnen schrijven met behulp van deze tool.

Dit leidde wel tot een interessante vraag: is het zinvol om interfaces te testen? Mijn mentor vond van niet, dus deze tests zijn later verwijderd.

Prompt engineering

Donderdag ging ik verder met het schrijven van tests, maar besteedde ook aandacht aan het verbeteren van mijn prompts voor de AI-tool. Ik merkte dat sommige prompts ruimte voor interpretatie lieten, wat tot onduidelijke resultaten leidde. Door mijn prompts te verfijnen, kon ik de kwaliteit van de gegenereerde code significant verbeteren.

Onverwachte terugkeer naar Timotheus

Vrijdag dacht ik weer de hele dag tests te gaan schrijven, maar Timotheus kwam nog eens om de hoek kijken. Er waren specifieke verzoeken om het logo en social media links in de e-mails te integreren die na het invullen van een test werden verzonden. Ook bleek één van de testen, de stresstest, nog steeds niet correct te werken.

E-mail templates verbeteren

Ik begon met het toevoegen van de banner in beide e-mails, gevolgd door het integreren van de logo's. Omdat de afbeeldingen verschillende dimensies hadden, moest ik mijn designvaardigheden

inzetten en heb ik in Figma de afbeeldingen aangepast.

Na enkele iteraties en tests realiseerde ik me dat Outlook en sommige andere e-mailproviders CSS anders interpreteren en met tabellen werken.

Stresstest fixen

Vervolgens heb ik de stresstest geanalyseerd. Mijn vermoeden dat randgevallen zoals gelijke scores het probleem veroorzaakten, bleek correct. Na het implementeren van een fix en het uitgebreid testen, bleek het probleem opgelost.

Hardwareupgrade en diepgaand begrip

Aan het einde van de week ontving ik nieuwe hardware: een kabel om twee schermen te kunnen gebruiken en een headset voor calls. Deze upgrade zal mijn workflow verder verbeteren.

Mijn mentor benadrukte het belang van het echt begrijpen van alle tests en de achterliggende code, in plaats van alleen maar AI-gegenererde tests te gebruiken. Dit is een belangrijke les voor mijn verdere ontwikkeling als software engineer.

Conclusie

Week 5 was opnieuw een week vol technische uitdagingen, maar ook met belangrijke leermomenten over de balans tussen productiviteit en diepgaand begrip. De combinatie van CI/CD workflows, unit testing en het gebruik van AI-ondersteunde ontwikkeltools heeft me veel nieuwe inzichten gegeven.

Volgende week ga ik verder met het dieper begrijpen van de codebase en het verbeteren van mijn unit tests, met een goede balans tussen AI-ondersteuning en eigen analyse.

Comments

0 reactions



0 comments

Write Preview Aa

Sign in to comment

M+

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#)[Blog](#)[Over Mij](#)[Contact](#)[Log In](#) [Terug](#)

[CI/CD](#)[VSCodeExtensions](#)[BusinessNetworking](#)

Week 6: Van Trailblazer Tot Beurservaring



Ian Mondelaers

May 12, 2025

Share:



Een week vol technische uitdagingen, nieuwe projecten en waardevolle netwerkervaringen

Technische Uitdagingen Met Trailblazer

Mijn week begon met een onverwachte uitdaging die mijn mentor me voorschoteldde: het updaten van de Trailblazer library in het Core.All project. Dit was geen eenvoudige taak gezien de complexiteit van de vele lagen en het feit dat het een third-party software betreft. Mijn mentor verwachtte eerlijk gezegd niet dat ik het zou kunnen oplossen.

Het proces bestond uit meerdere stappen:

1. De vernieuwde externe library lokaal bouwen
2. Nuget packages genereren

3. Deze implementeren in het Core.All project
4. De CI/CD pipeline (Gitea Actions) aanpassen

Tijdens dit proces leerde ik enkele waardevolle lessen over CI/CD workflows, waaronder het gebruik van wildcards in actions (`**` in Gitea's syntax) en het configureren van acties om op specifieke tijden te draaien met UTC timezone-instellingen.

Een belangrijk inzicht kwam toen ik merkte dat er merge conflicts ontstonden door verschillen tussen mijn branch en de main branch. Dit benadrukte het belang van regelmatig synchroniseren met de hoofdontwikkelingstak:

"Dit is zeer spijtig, maar hier leer ik uit om vaker de main branch binnen te trekken en te letten op wat andere developers doen en aan het werken zijn."

Billit Facturatie-Optimalisatie

Een interessante zijstap was het helpen van Anna met het facturatiesysteem Billit. Ze wilde graag JavaScript-functionaliteit toevoegen aan de factuurtemplate om sommen automatisch te berekenen, plus wat visuele verbeteringen doorvoeren.

Na grondig onderzoek ontdekte ik dat de Billit-omgeving helaas JavaScript-code wegfiltert uit

templates, waardoor de gewenste functionaliteit niet mogelijk was. Wel kon ik de CSS-styling succesvol optimaliseren om de facturen visueel aantrekkelijker te maken. Een mooi voorbeeld van hoe je binnen de beperkingen van een systeem toch waarde kunt toevoegen.

VSCODE Plugin Ontwikkeling

Een groot deel van mijn week besteedde ik aan een bijzonder uitdagend project: het ontwikkelen van een VSCode-extensie voor HolonCom. Dit betreft een "knowledge graph generator" die essentieel functioneert als een geïntegreerd woordenboek voor bedrijfsspecifieke terminologie.

Het ontwikkelproces begon met het grondig doornemen van de documentatie:

- Extension Test Runner voor het testen van extensies
- Het ``.vscodeignore`` bestand voor optimale extensiegrootte
- Best practices voor het bijhouden van een changelog

Na enkele structurele uitdagingen kreeg ik een werkende basis-plugin. Vervolgens ben ik begonnen met het implementeren van een hover-functionaliteit die informatie weergeeft wanneer je met de cursor over een woord zweeft. De structuur van de code heb ik verbeterd door separation of concerns toe te passen, wat de onderhoudbaarheid en leesbaarheid van de code ten goede komt.

Business Experience: Netwerken Op De Beurs

Donderdag was een bijzondere dag. Terwijl mijn collega's al vroeg op de B2B-beurs stonden, werkte ik 's ochtends nog aan de plugin en voegde ik in de namiddag aan bij het evenement.

Aanvankelijk voelde ik me wat overweldigd door de professionele setting, maar dankzij de aanmoediging van mijn collega's durfde ik diverse standjes te bezoeken en gesprekken aan te knopen. Dit leverde waardevolle connecties op met onder andere:

- Een SEO-expert die me inzichten gaf in de technische aspecten van zoekmachineoptimalisatie
- Een bedrijf gespecialiseerd in refurbished elektronica, wat aansloot bij mijn persoonlijke interesse
- Een ondernemster die haar eigen bedrijf "Blij Bakkes" startte als student en me haar contactgegevens gaf voor verdere vragen over ondernemerschap

Deze ervaring was een perfecte aanvulling op mijn technische stage en bood me waardevolle soft skills en netwerk mogelijkheden.

Troubleshooting En Samenwerking

Vrijdag stond in het teken van probleemoplossing en samenwerking. Dries contacteerde me met een

dependency-probleem in een project dat gebruik maakte van een door mij gemaakte nuget package. Samen hebben we het probleem geanalyseerd en uiteindelijk opgelost, wat een goede oefening was in cross-team samenwerking en technisch debuggen.

De week sloot ik af met het afronden van het Timotheus-project, waarbij ik links optimaliseerde en de copywriting verbeterde. Door het geplande online zetten op maandag moest alles perfect in orde zijn. Ik besteedde extra aandacht aan de details en voegde zelfs emoji's toe om bepaalde onderdelen levendiger te maken.

Conclusie

Week 6 was gevuld met diverse technische uitdagingen die mijn vaardigheden als software engineer hebben aangescherpt. Van complexe library-updates en CI/CD-configuratie tot plugin-ontwikkeling en praktisch probleemoplossen - elke dag bood nieuwe leerkansen.

De beurservaring vormde een waardevolle aanvulling op mijn technische ontwikkeling, en toonde hoe belangrijk interpersoonlijke vaardigheden zijn in een IT-carrière. De week eindigde positief met een geslaagde demo van mijn plugin-voortgang aan mijn mentor, wat bevestigde dat ik op de juiste weg zit.

Comments

0 reactions



0 comments

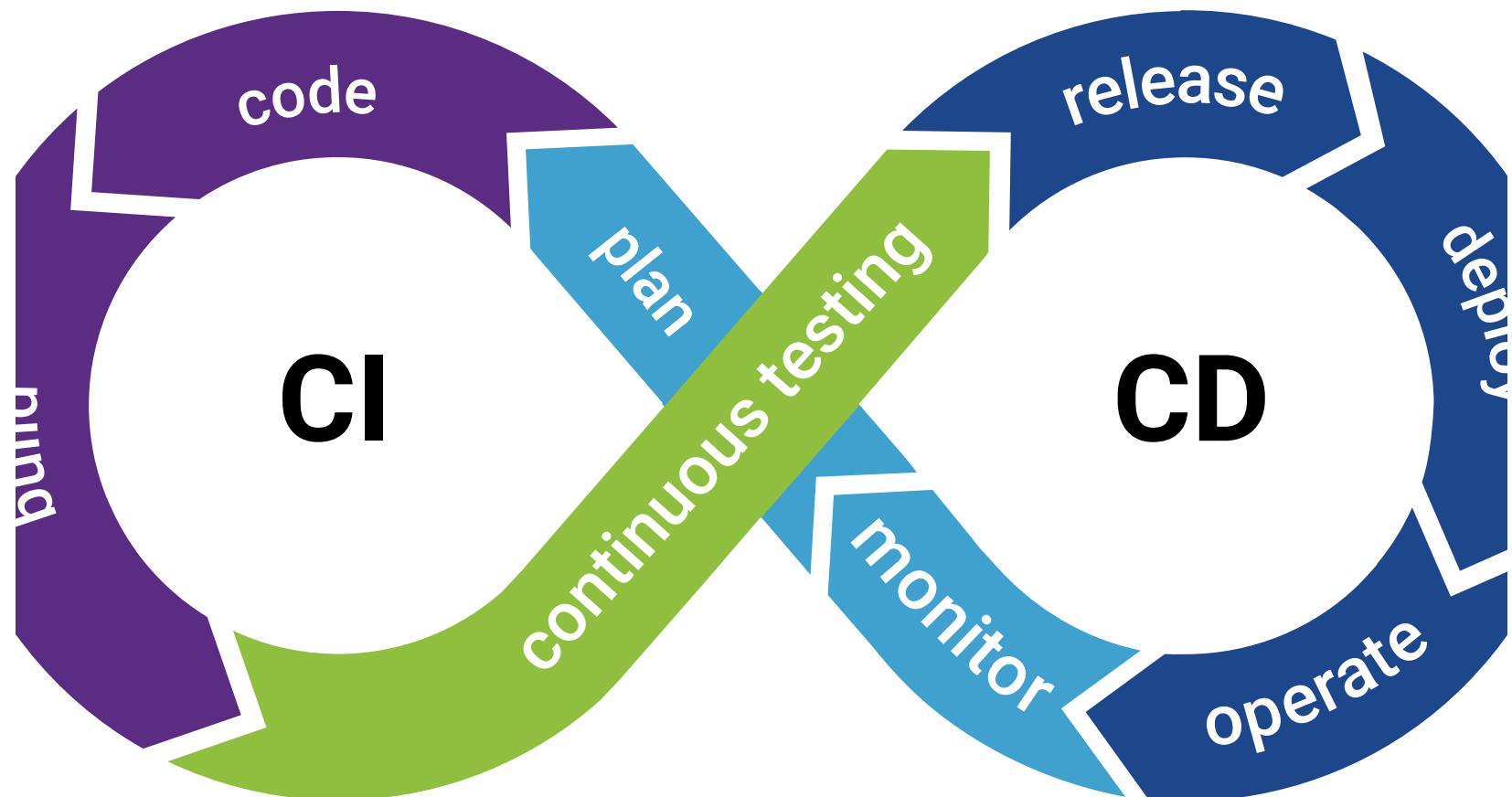
Write Preview Aa

Sign in to comment

M+

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#)[Blog](#)[Over Mij](#)[Contact](#)[Log In](#) [Terug](#)

[VSCodeExtensions](#)[Debugging](#)[PersonalGrowth](#)

Week 7: Docker, CI/CD en Plugin Development in Volle Gang

**Ian Mondelaers**

May 12, 2025

Share:



Deze week stond in het teken van containerisatie, plugin-verbeteringen en DevOps-uitdagingen

VSCODE PLUGIN: Multiplatform Compatibiliteit en Feature Uitbreiding

Een van mijn eerste opdrachten deze week was het uitbreiden van de VSCode Knowledge Graph plugin die ik vorige week ontwikkelde. Een belangrijk doel was om de plugin niet alleen in VSCode maar ook in Cursor IDE werkend te krijgen.

Het probleem bleek te zitten in een versieconflict – de plugin was opgebouwd voor VSCode 1.98.0 terwijl mijn Cursor IDE werkte op versie 1.96.2. Door de version requirements in het `'package.json'` bestand aan te passen, kon ik de plugin succesvol draaiend krijgen in beide omgevingen. Hoewel een

schijnbaar kleine aanpassing, was dit een belangrijke stap in het verbeteren van de compatibiliteit en gebruiksvriendelijkheid van de plugin.

Na deze bugfix heb ik een gestructureerde roadmap opgesteld voor verdere ontwikkeling:

Roadmap Knowledge Graph Plugin:

0. Ondersteuning voor meerdere bestandsformaten
1. Verbetering van conceptherkenning en formatting
2. Conceptnormalisatie en synoniemen

De implementatie verliep voorspoedig met verschillende belangrijke verbeteringen:

- Support voor YAML, Markdown en TXT bestanden
- Een WYSIWYG interface met formulier voor nieuwe concepten/relaties
- Automatische bestandstype-detectie
- Caching en incrementele parsing voor betere performance
- Nieuwe manieren om relaties tussen concepten aan te duiden, zoals ...

Om het project professioneel af te ronden, heb ik ook de nodige aandacht besteed aan een MIT-

licentie:

"Ik merkte dat mijn project geen licentie had en wou hier graag over bijleren. Ik vond de website opensource.org en koos voor de MIT-licentie, die populair en veel gebruikt is in de open-source community."

Van DLLs naar NuGet: De Overgang naar Moderne Package Management

Een interessante uitdaging was het overzetten van het Rabbitrest-project van lokale DLL-references naar NuGet packages. Dit was een belangrijk migratieproject dat de codebase moderner en beter onderhoudbaar moest maken.

Het proces omvatte:

1. Het opzetten van een `nuget.config` bestand
2. Configuratie van `global.props` voor projectbrede instellingen
3. Aanpassing van projectbestanden voor NuGet-compatibiliteit
4. Het opruimen van de `lib` -folder, waarbij niet-gebruikte DLLs werden verwijderd

Voor projecten waarvoor nog geen NuGet packages beschikbaar waren, behield ik voorlopig de DLL-references – een pragmatische aanpak om de overgang geleidelijk te laten verlopen zonder de bestaande functionaliteit te verstoren.

Docker Containerization: CI/CD Workflows Automatiseren

Een groot deel van mijn week werd in beslag genomen door het opzetten van Docker containers en het integreren hiervan in de CI/CD pipeline via Gitea Actions. Dit bleek een uitdagende, maar leerzame ervaring.

Initieel liep ik tegen een obstakel aan toen bleek dat mijn virtuele machine niet goed was ingesteld voor Docker Desktop:

"Het probleem lag bij 'nested virtualization' die niet aanstond in de BIOS van de virtuele machine. Ik heb alle nodige Windows features aangezet en WSL2 geïnstalleerd, maar de VM-platform ondersteunde geen nested virtualization."

Om toch voortgang te boeken, heb ik het project gekopieerd naar een fysieke machine waarop ik Docker wel kon installeren. Daar leerde ik enkele belangrijke Docker best practices:

- Variabelen moeten correct geconfigureerd worden via `.**env**` bestanden of omgevingsvariabelen

- Docker image names mogen geen hoofdletters bevatten
- Relatieve paden worden anders geïnterpreteerd binnen de Dockerfile context

De grootste uitdaging was het opzetten van een geautomatiseerde workflow die Docker images zou bouwen en deployen naar de Gitea registry. Dit vereiste aandacht voor:

- Correcte padverwijzingen vanuit verschillende perspectieven (Dockerfile vs. Action)
- NuGet package resolution in de CI/CD omgeving
- Efficiënte build processen (debug vs. release mode)
- Timeout-problemen tijdens het cachen van builds

Na verschillende iteraties en opzoekwerk over Github Actions, kon ik een schone en functionele workflow implementeren:

"Na wat tijd en opzoekingswerk op Github, heb ik wat gelijkwaardige actions gevonden die ik kan gebruiken als basis. Zo kan de Dockerfile zich nu op verschillende plaatsen bevinden en heb ik de hacky manier voor het vinden van NuGets eruit kunnen halen, waardoor de action veel cleaner is."

Command Worker: Complexe Containerisatie

Een extra uitdaging kwam aan het eind van de week met het containeriseren van de Command Worker applicatie – een complex project met zowel Blazor Server, Blazor Client als API componenten.

Deze taak bleek extra complex vanwege de omvang en structuur van het project:

"Dit is een pittige opdracht aangezien het zo'n groot project is met zowel een Blazor server als client en een API."

Om deze complexiteit te beheersen, nam ik verschillende stappen:

1. Toevoegen van gezondheidscontroles (health checks) aan de API voor monitoring
2. Creëren van een aparte solution zonder testprojecten (Docker en testomgevingen bleken niet goed samen te werken)
3. Refactoren van projectbestanden voor betere compatibiliteit met Docker en CI/CD workflows

Het Belang van Test Maintenance

Een belangrijk leermoment kwam toen ik voor het Timotheus-project eerder geschreven testen

moest reactiveren:

"Door zoveel code veranderingen waren deze niet meer werkend met de huidige codebase. Mijn mentor noemde dit een 'los eindje' en vertelde me dat dit een veel voorkomend probleem is binnen IT waar hij niet van houdt."

Dit benadrukte het belang van het up-to-date houden van testen, zelfs tijdens snelle ontwikkelcycli. De testen helpen uiteindelijk om alle mogelijke testresultaten te valideren en in de `bin` folder op te slaan voor verificatie.

Conclusie en Reflectie

Week 7 kenmerkte zich door diepgaand werk aan verschillende DevOps-aspecten: containerisatie, CI/CD automatisering, en package management-modernisering. Deze vaardigheden vormen een essentieel onderdeel van moderne software engineering en geven me waardevolle ervaring voor toekomstige projecten.

De uitdagingen met Docker en CI/CD pipelines hebben me geleerd hoe belangrijk het is om systematisch problemen aan te pakken en geduldig te zijn bij het debuggen van complexe buildprocessen. Tegelijkertijd heeft de plugin-ontwikkeling mijn front-end en development-tool vaardigheden verder aangescherpt.

Aan het eind van de week stond er een langdurige action te draaien – een mooi symbool voor hoe software development vaak werkt: je zet processen in gang en wacht geduldig op resultaten, klaar om volgende week verder te bouwen op wat je hebt bereikt.

Comments

0 reactions



0 comments

Write Preview Aa

Sign in to comment

M

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#)[Blog](#)[Over Mij](#)[Contact](#)[Log In](#) [Terug](#)

[VSCodeExtensions](#)[SoftwareArchitecture](#)[WebDevelopment](#)

Week 8: Persoonlijke groei, Docker en Klantprojecten

**Ian Mondelaers**

May 13, 2025

Share:



Deze week stond in het teken van uitdagende refactorings, mijn tussentijdse evaluatie en interessante gesprekken over professionele groei. Daarnaast kreeg ik de kans om voor het eerst "zelfstandig" een klantproject op te zetten.

Debugging en CI/CD pipeline optimalisatie

Maandag begon ik met het oplossen van errors in de Command Worker. De vorige week had ik al gewerkt aan het opzetten van GitHub Actions voor automatische Docker image builds, en tot mijn vreugde werkte dit. Mijn eerste succesvolle Docker image is een feit!

De debugging bleek echter complexer dan verwacht. De Command Worker is sterk afhankelijk van het HolonCom.Core.All project, waardoor problemen in de ene codebase vaak doorwerken in de andere. Een specifieke uitdaging was dat de build online faalde terwijl deze lokaal wel werkte. Na

analyse bleek dit te komen door:

1. Typfouten in packagename
2. Ontbrekende packages
3. Lokale wijzigingen die nog niet gereflecteerd waren in de NuGet packages

De oplossing bestond uit twee fasen:

- Eerst de build met lokale projectreferenties laten werken
- Daarna overschakelen naar NuGet packages

Dit proces hielp me beter begrijpen hoe dependencybeheer werkt in grotere .NET-oplossingen.

Tussentijdse evaluatie en persoonlijke groei

Dinsdag vond mijn tussentijdse evaluatie plaats met mijn stagebegeleider. Voorafgaand had ik een voorbereidend gesprek met mijn mentor. Hij gaf me waardevolle feedback over mijn stageperiode tot nu toe: over het algemeen positief, maar met het advies om vaker de "waarom"-vraag te stellen. Hij wil meer van "mij" zien door kritischer te zijn en dieper te graven.

Een interessante metafoor die hij deelde: "De arend vliegt alleen, en het doet enigszins pijn om niet meer met de mussen te kunnen spelen." Dit reflecteert de realiteit van professionele groei.

We bespraken ook concepten als "associatie" en "dissociatie" in relatie tot zelfreflectie en persoonlijke ontwikkeling. Deze gesprekken helpen me om niet alleen technisch te groeien, maar ook professioneel.

Een kleine les in professionele etiquette: toen mijn stagebegeleider arriveerde, had ik hem geen koffie aangeboden omdat ik niet wist hoe de koffiemachine werkte. Mijn mentor merkte op dat juist dit soort details het verschil maken tussen goed en uitstekend. Beter proberen en falen dan helemaal niet proberen - een les die verder reikt dan alleen koffie aanbieden.

De evaluatie zelf was zeer positief. Ik kijk uit naar de schriftelijke feedback om specifieke verbeterpunten te identificeren.

Docker en versioning challenges

Woensdag en donderdag stond Docker centraal. Ik worstelde met versioning issues in onze GitHub Actions pipeline. Bij het analyseren van de error logs bleek dat de oudste stabiele versies van bepaalde packages versieconflicten veroorzaakten.

Ik onderzocht oplossingsrichtingen zoals:

- Het gebruik van `directory.packages.props` voor centrale versioning
- Het optimaliseren van de `nuget.config` voor package source mapping

De `nuget.config` aanpassingen waren succesvol, maar de eerste optie bleek te ingrijpend voor de huidige projectstatus, aangezien alle versies (inclusief externe NuGets) verwijderd zouden moeten worden.

Nieuwe projecten en klantwerk

Halverwege de week kreeg ik een nieuwe opdracht voor een klant die een statische website wenste. Na research van verschillende oplossingen (Hugo, Jekyll) kozen we voor Ghost - een gebruiksvriendelijk CMS dat met markdown werkt.

Ik zette een Docker container op met een SQLite database voor de blog. Een uitdaging hierbij was het correct configureren van de database-persistence en het oplossen van SSL-issues bij het mappen van poorten. Dit veroorzaakte bugs in het admin panel, zoals niet-werkende previews.

Daarnaast werd ik betrokken bij een E-shop project voor snelle bugfixing en refactoring. Hier leerde ik meer over:

- Het oplossen van solution errors

- Het updaten van project files
- Het oplossen van packagereferentie issues in .NET3-projecten (waarbij packages in een packages.config file zitten in plaats van in de project files)

Het was interessant om te zien hoe je bij oudere .NET-projecten soms de project files handmatig in een teksteditor moet aanpassen omdat Visual Studio niet alle details toont.

Commerciële en persoonlijke ontwikkeling

Een waardevol moment deze week was het gesprek met mijn mentor over mogelijkheden als student-zelfstandige na mijn stage. We bespraken:

- Statuut als zelfstandige: belastingaspecten en sociale zekerheid
- Het verschil tussen "natuurlijk persoon" en "rechtspersoon"
- Financiële planning en investeringen

Dit gesprek opende mijn ogen voor de zakelijke kant van softwareontwikkeling en gaf me stof tot nadenken over mijn toekomstige carrièrepad.

Conclusie

Week 8 was een mix van technische uitdagingen, professionele groei en nieuwe projectervaringen. De combinatie van dagelijkse ontwikkeltaken en de bredere gesprekken over professionele ontwikkeling maken deze stage tot een waardevolle ervaring.

Ik kijk uit naar volgende week, waarin ik hoop de Ghost-implementatie af te ronden en verder te werken aan de Command Worker optimalisatie.

Comments

0 reactions



0 comments

Write

Preview

Aa

Sign in to comment



Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#) [Blog](#) [Over Mij](#) [Contact](#) [Log In](#) [Terug](#)

ghost

[WebDevelopment](#)[UnitTesting](#)[Debugging](#)[Docker](#)

Week 9: Projectrefactoring, NuGet-optimalisaties en Ghost CMS

**Ian Mondelaers**

May 13, 2025

Share:



Deze week stond in het teken van diepgaande refactoring van projectstructuren, het opzetten van een efficiënte NuGet workflow en verdere ontwikkeling van het Ghost CMS project voor een klant. Ook moest ik verschillende oplossingen ontwikkelen voor onverwachte bouwfouten in complexe solution-structuren.

Ghost CMS projectontwikkeling

Maandag begon vroeg met verdere ontwikkeling aan het Ghost CMS project. Bij het analyseren van de containerlogboeken ontdekte ik dat er nog steeds pogingen werden gedaan om verbinding te maken met een MySQL-server, ondanks dat ik deze configuratie uit de Docker Compose file had verwijderd. De oplossing lag in het expliciet configureren van Ghost om een SQLite database te gebruiken:

```
# Database configuration  
database_client: sqlite3  
database_connection_filename: /var/lib/ghost/content/data/ghost.db
```

Dit verhielp het probleem waarbij de container bleef herstarten vanwege mislukte verbindingspogingen met een niet-bestaaende MySQL-server.

Later in de week maakte ik twee versies van de Docker Compose-configuratie:

1. Een versie met SQLite voor eenvoudige implementaties
2. Een variant met MySQL voor productieomgevingen met hogere prestatie-eisen

Daarnaast creëerde ik automatiseringsscripts (zowel PowerShell als Bash) om de juiste permissies toe te kennen aan de content-map, onafhankelijk van het besturingssysteem waarop de container draait. De volledige configuratie werd afgerond met een Nginx configuratiebestand voor het opzetten van een reverse proxy met SSL-certificaatondersteuning voor domeinverbindingen.

E-shop refactoring en modernisering

Een groot deel van deze week heb ik besteed aan het refactoren van de E-shop codebase. Het

project bevatte diverse uitdagingen:

- Problemen met constructors en overervende klassen die tijdens eerdere refactoring niet correct waren bijgewerkt
- Verouderde project SDK-versies die toekomstige onderhoudsproblemen zouden kunnen veroorzaken
- Een complexe afhankelijkheidsstructuur tussen de verschillende projecten

Mijn eerste focus lag op het implementeren van een switch-mechanisme om eenvoudig te kunnen schakelen tussen lokale projectreferenties en NuGet-packages. Dit was een strategische beslissing om de ontwikkelworkflow te verbeteren:

- Tijdens actieve ontwikkeling zijn lokale projectreferenties handig voor directe debugging
- Voor stabiele builds en deployment zijn NuGet-packages de voorkeur

Het was interessant om te leren dat je voor .NET Framework projecten de projectbestanden vaak handmatig moet bewerken in een teksteditor, omdat Visual Studio niet alle details zichtbaar maakt in de GUI. Een waardevolle technische vaardigheid die ik nu heb opgedaan.

SDK-upgrade uitdagingen

Een van mijn meer ambitieuze taken was het upgraden van de SDK-versie voor verschillende projecten om deze toekomstbestendig te maken. Hoewel ik hier niet veel ervaring mee had, besloot ik de uitdaging aan te gaan.

Na mijn initiële refactoring had ik een versie die succesvol kon bouwen, maar er bleken onverwachte problemen te zijn met een target-bestand dat verantwoordelijk is voor het genereren van assembly-DLL's. Deze werd niet correct geïmporteerd of gebruikt na de SDK-upgrade.

Mijn mentor was verrast en geïnteresseerd in mijn poging om dit aan te pakken:

"Hij vond het wel interessant en had niet verwacht dat ik dit zou doen of dat dit me (deels) zou lukken."

Vanwege tijdsbeperkingen hebben we besloten deze specifieke refactoring tijdelijk te pauzeren, maar het was een leerzame ervaring in het begrijpen van de complexiteit van .NET project upgrades.

NuGet workflow-optimalisatie

Een van de meest uitdagende aspecten van deze week was het optimaliseren van de NuGet-workflowstrategie. De hoofduitdaging was het valideren van welke packages werden gebruikt (lokaal versus online) en hoe we een betrouwbare schakelaar konden implementeren.

Mijn initiële strategie was het toevoegen van een opzettelijke fout (division by zero) in lokale code om te controleren of de correcte packages werden gebruikt:

- Als tests slaagden, werden online packages gebruikt
- Als tests faalden met de division by zero exception, werden lokale packages gebruikt

Nadat ik merkte dat de eenvoudige switch niet werkte zoals verwacht, onderzocht ik het gebruik van MSBuild-variabelen in de NuGet.config. Het idee was om voorwaardelijke logica toe te voegen aan de pakketbronverwijzingen.

Ik richtte een lokale NuGet-registry in door:

1. Global.props (fungerend als Directory.Build.props) te configureren met een pad voor package-output
2. PackageOnBuild op true te zetten om automatisch packages te genereren
3. Conditionele verwijzingen naar de lokale registry toe te voegen

Helaas ontdekte ik dat MSBuild-variabelen niet op tijd worden geladen voor de NuGet.config. De uiteindelijke oplossing was pragmatischer: beide registries toevoegen aan de config en handmatig commentaar toevoegen aan degene die niet gebruikt moet worden. Een belangrijk aandachtspunt hierbij is dat de bin/obj-mappen moeten worden opgeschoond na het wisselen tussen registries.

Testprojecten moderniseren

Een andere belangrijke refactoring was het omzetten van bestaande MSTest-projecten naar xUnit. Dit biedt moderne testfunctionaliteit en betere integratie met de huidige ontwikkelomgeving. Hierbij liep ik tegen een interessant probleem aan:

Het probleem was dat ik sommige testklassen nog niet had omgezet naar xUnit code, maar wel de project files refactored had naar xUnit. Hierdoor build de solution niet meer.

Deze ervaring leerde me een waardevolle les: zorg ervoor dat de migratie van testcode en de bijbehorende projectconfiguratie synchroon verlopen om onverwachte bouwfouten te voorkomen.

Documentatie en kennisdeling

Tegen het einde van de week heb ik aanzienlijke tijd besteed aan het verbeteren van de documentatie, met name door README-bestanden aan te vullen of te creëren voor de Core.All-projecten. Goede documentatie is cruciaal voor de overdraagbaarheid van kennis, vooral in complexe projectstructuren zoals deze.

Na een nacht reflectie kon ik veel verbeteringen aanbrengen in de documentatie die ik eerder had geschreven, wat het belang onderstreept van het laten bezinken van ideeën en periodieke herziening van technische documentatie.

Samenwerking en probleemoplossing

Deze week was ook een goede ervaring in teamwerk. Toen een collega-ontwikkelaar, Dries, meldde dat de Command Worker niet meer kon bouwen, gingen we samen troubleshooten:

We gingen in call en ik merkte dat een project dat recent is toegevoegd nog niet refactored was om de switch te gebruiken en dit gaf problemen met dubbele referenties (nugets - projects).

Ik hielp hem ook door projecten van de Command Worker over te zetten naar Core.All en de nodige refactoring uit te voeren. Dit soort directe samenwerking is onzettend waardevol voor kennisuitwisseling en efficiënte probleemoplossing.

Conclusie

Week 9 was technisch uitdagend en zeer leerzaam. Ik heb waardevolle ervaring opgedaan met:

- Het configureren van container gebaseerde applicaties
- Complexe projectrefactoring en SDK-upgrades
- NuGet packagebeheer en lokale/online registrystrategieën

- Het moderniseren van testprojectstructuren
- Het schrijven van effectieve technische documentatie

Deze ervaringen hebben me een dieper inzicht gegeven in softwareontwikkeling op enterpriseniveau, waarbij het beheren van complexe afhankelijkheden en het onderhouden van legacy-code vaak een cruciale rol speelt.

Comments

0 reactions



0 comments

Write

Preview

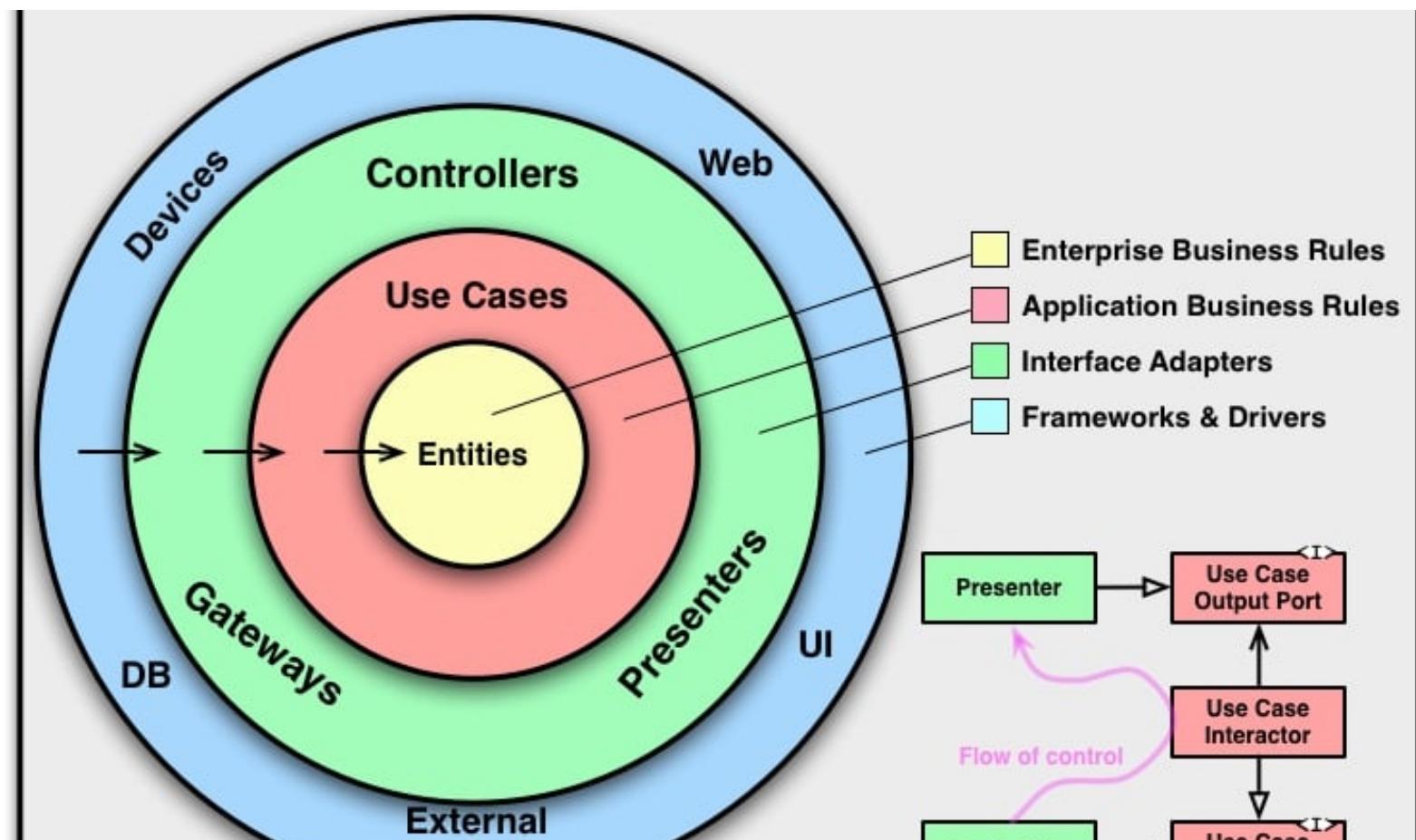
Aa

Sign in to comment



Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[← Terug](#)

[VSCodeExtensions](#)[WebDevelopment](#)[Deployment](#)[Docker](#)

Week 10: Tussen Test Automatie en Telefonie



Ian Mondelaers

May 14, 2025

Share:



Deze week stond in het teken van diverse projecten en nieuwe verantwoordelijkheden. Van het ontwikkelen van robuuste teststrategieën tot het voorbereiden op een nieuwe uitdaging in de vorm van telefonische ondersteuning - week 10 was gevuld met leermomenten die mijn technische vaardigheden en professionele ontwikkeling verder hebben versterkt.

Testautomatisering en CI/CD Verbeteringen

Een groot deel van mijn week is besteed aan het verbeteren van de testinfrastructuur binnen het Core.All project. Ik heb een uitgebreide test suite ontwikkeld en een efficiënte workflow opgezet die bij elke code push automatisch alle tests uitvoert. Een van de meest bevredigende verbeteringen was het toevoegen van een status badge aan de README van de repository, die real-time de status van de tests weergeeft.

Deze test workflow leverde onmiddellijk waarde op door inconsistenties te identificeren in de manier

waarop line endings werden verwerkt op Unix-gebaseerde systemen. Daarnaast ontdekte ik een bug in een methode die raw generic types niet correct herkende. Dit onderstreept nog maar eens het belang van een cross-platform testbenadering.

Verdieping in Vertical Architecture

Een bijzonder waardevol moment deze week was een diepgaand gesprek met mijn mentor over "vertical architecture". In tegenstelling tot traditionele architecturen zoals MVC, waarin code wordt georganiseerd op basis van technische aspecten, benadrukt vertical architecture het organiseren van code rond business functies of "slices".

De kern van deze benadering is Domain Driven Development, waarbij de structuur van de code het businessdomein reflecteert in plaats van technische overwegingen. Voor een To-Do applicatie betekent dit bijvoorbeeld:

```
ToDoApp/
  └── Features/
      └── CreateToDo/
          ├── CreateToDoController.cs
          ├── CreateToDoHandler.cs
          └── CreateToDoValidator.cs
      └── CompleteToDo/
```

```
|   |   └─ CompleteToDoController.cs  
|   |   └─ CompleteToDoHandler.cs  
|   └─ CompleteToDoValidator.cs  
└─ ...
```

Deze architectuur zorgt voor een betere scheiding van concerns en maakt de code beter onderhoudbaar en uitbreidbaar. Het was fascinerend om te zien hoe deze principes worden toegepast in de projecten van HolonCom.

Nieuwe Verantwoordelijkheid: Telefonie Ondersteuning

Een nieuwe uitdaging die deze week op mijn pad kwam, was de voorbereiding op het overnemen van de telefonische ondersteuning tijdens de vakantie van mijn mentor en Anna. Hiervoor werd 3CX geïnstalleerd, een telefonie-oplossing.

Om mezelf voor te bereiden op deze verantwoordelijkheid, heb ik:

- Een draaiboek opgesteld met scripts voor verschillende scenario's
- Een gestructureerd template gemaakt voor het verzamelen van essentiële informatie tijdens gesprekken

- Mock call uitgevoerd met mijn mentor om mijn aanpak te verfijnen

De feedback was overwegend positief, met enkele nuttige tips voor het verbeteren van mijn intonatie.

De belangrijkste taak zal zijn om als "doorgeefluik" te functioneren, waarbij ik alleen bij urgente kwesties contact opneem met mijn mentor of Anna.

Ghost Project en Content Management

Naast mijn werkzaamheden aan tests, heb ik ook vooruitgang geboekt met het Ghost project. Dit betrof voornamelijk het opzetten van de website structuur en het toevoegen van content zoals links, pagina's, tags en instellingen.

Na een demo aan mijn mentor kreeg ik waardevol feedback:

- Meer experimenteren met de Pages-functionaliteit
- De Explore-tab verder onderzoeken
- Onderzoeken of er een API beschikbaar is voor het creëren van blogposts via externe programma's

Het Ghost project is inmiddels online gezet via Portainer, wat betekent dat ik er gemakkelijker aan kan werken en mijn wijzigingen direct live kan testen.

Onderzoek naar Nieuwe Tooling

Tijdens mijn werkzaamheden aan de testprojecten ben ik tegen enkele interessante tools aangelopen:

CodeCov

Ik heb CodeCov onderzocht, een tool die het monitoren en verbeteren van test coverage vereenvoudigt. Hoewel de tool veelbelovend is, bleek integratie met Gitea helaas niet mogelijk. Dit ondanks verschillende pogingen om dit te forceren via custom workflows.

Scriban

Een andere ontdekking was Scriban, een lichtgewicht templating engine met een syntax die dicht bij C# ligt. De mogelijkheden hiervan heb ik gedocumenteerd in een README voor toekomstig gebruik.

Vooruitblik en Planning

Aan het einde van de week heb ik een pragmatische planning opgesteld voor de periode dat mijn mentor afwezig zal zijn. Dit omvat:

- Een gestructureerd systeem voor het bijhouden van call logs
- Een stappenplan voor mijn werkzaamheden in de komende week
- Duidelijke communicatielijnen, inclusief het telefoonnummer van mijn mentor voor nood gevallen

Conclusie

Week 10 was een week van technische verdieping en nieuwe verantwoordelijkheden. Het verbeteren van de testinfrastructuur, het begrijpen van vertical architecture, en het voorbereiden op telefonische ondersteuning hebben me waardevolle inzichten gegeven in zowel technische als professionele aspecten van softwareontwikkeling.

De combinatie van hands-on codering en conceptuele discussies met mijn mentor blijft een uitstekende leerschool. Ik kijk uit naar de uitdaging om volgende week meer zelfstandig te werken tijdens de afwezigheid van mijn mentor.

Comments

0 reactions



0 comments

Write

Preview

Aa

Sign in to comment



Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[← Terug](#)

[WebDevelopment](#)[Telefonie](#)[Docker](#)

week 11: Bugs, refactoring en projecten



Ian Mondelaers

May 14, 2025

Share:



Debug-avonturen met de KnowledgeGraph plugin

Deze week begon met het verder werken aan de KnowledgeGraph plugin - een Visual Studio Code extensie die bij HolonCom belangrijk blijkt voor kennisbeheer binnen projecten. Maandag heb ik eerst de documentatie grondig verbeterd: overlappende informatie verwijderd, verouderde delen bijgewerkt en verschillende markdown bestanden een duidelijker doel gegeven (changelog, readme, roadmap).

Daarna ben ik systematisch door de codebase gegaan om verbeterpunten te identificeren. Ik heb enkele bevindingen gedaan die directe aandacht nodig hadden:

- Type validatie issues waarbij `as unknown as occurrences` casts werden gebruikt - een patroon dat tot runtime errors kan leiden
- Asynchrone initialisatieproblemen die potentiële race conditions veroorzaakten

- Diverse kleinere problemen rondom data-toegang, UX/UI elementen en algemene code kwaliteit

Tijdens het testen ontdekte ik een merkwaardig probleem: de debug logs gaven aan dat er geen links of data werd gemapped bij activatie van de plugin. Dit bleek uiteindelijk een symptoom van een dieper probleem te zijn. Na uren van debugging ontdekte ik dat de `dist`-map (waar de gecompileerde code zou moeten staan) een verouderde compilatie bevatte. Dit leidde tot een reeks onverwachte problemen - mijn nieuwe features en fixes waren eigenlijk nooit echt getest!

```
// Voorbeeld van een probleem dat ik oploste:  
// Oude code – problematische type casting  
const occurrences = someData as unknown as occurrences;  
  
// Nieuwe code – type-veilige benadering  
if (isOccurrences(someData)) {  
    const occurrences = someData;  
    // Verdere verwerking...  
}
```

Dinsdag dook ik dieper in de debugproblematiek. Wat ik leerde bevestigde iets waar ik al langer van overtuigd was: debugging is echt een onderschatte vaardigheid in software ontwikkeling. Na het verwijderen van de `dist`- en `out`-folders en het aanpassen van configuraties, werkte de plugin nog steeds niet.

De grootste doorbraak kwam toen ik ontdekte dat het probleem bij de YAML parser lag. De nieuw geïmplementeerde library was te strikt in vergelijking met de oude parser. Na een grondige refactoring van de parser werkte de ondersteuning voor YAML files weer. Als bonus heb ik ook de hover-functionaliteit verbeterd, inclusief de opmaak van tekst in de hover UI en een bug opgelost waardoor relation types geen descriptions toonden.

Structurele verbeteringen en nieuwe projecten

Woensdag splitste ik de logica van de link providers op, zodat elk type nu zijn eigen link provider heeft - een stap richting betere modulariteit en onderhoud van de code. Met de plugin nu in een stabiele staat, kon ik me richten op een ander project.

Mijn mentor had changes gepusht voor een project dat ik moest containerizen in Docker. Na eerst enkele refactors door te voeren om het project buildbaar te maken, heb ik de code geanalyseerd en een plan gemaakt. Ik creëerde de nodige Docker files en een GitHub Action, waarna ik verschillende iteraties nodig had om de actie succesvol te laten bouwen.

Er deden zich problemen voor met de NuGet packages. Een veelvoorkomend probleem was dat NuGet packages werden gemaakt bij elke push naar willekeurige branches, waardoor features werden gebruikt die nog niet compatibel waren:

```
# Aangepaste GitHub Action-configuratie:  
on:  
  push:  
    branches:  
      - develop  
      - release/*  
    # Zorgt dat NuGet packages alleen worden gemaakt van stabiele branches
```

Core libraries en NuGet optimalisatie

Op donderdag moest ik dieper duiken in het Core.All project vanwege problemen met NuGet packages. Sommige projects hadden geen NuGet packages, wat vaak wijst op projecten die ontbreken in de solution of niet correct geconfigureerd zijn voor package creatie.

Een hardnekkig probleem bleef zich voordoen met één specifieke package. Zelfs na het expliciet instellen van `IsPackable` op `true` werd de NuGet nog steeds niet correct gegeneerd in de GitHub Action. Uiteindelijk heb ik deze handmatig gepusht naar de online repository, wat weer nieuwe uitdagingen opleverde met het StreamDesk project.

Ik heb verbeteringen aangebracht in de `Global.props` file door een `IsPackable` variabele te introduceren en de lokale NuGet repository te optimaliseren:

```
<!-- Global.props aanpassingen -->
<PropertyGroup>
    <IsPackable Condition="'$(IsPackable)' == ''">true</IsPackable>
    <PackageOutputPath Condition="'$(Configuration)' == 'Debug'">$(MSBuildThisFileDirectory)..\` 
    <PackageOutputPath Condition="'$(Configuration)' == 'Release'">$(MSBuildThisFileDirectory).` 
</PropertyGroup>
```

Deze aanpassing zorgt ervoor dat packages nu in verschillende repositories komen afhankelijk van de build configuratie (debug of release), wat de ontwikkelingsworkflow verbetert.

Blog website en visuele identiteit

Verder werkte ik deze week aan een blog website voor een klant. Na contact met hem te hebben opgenomen, kon ik zijn bestaande WordPress posts exporteren. Dit maakte het migratieproces aanzienlijk eenvoudiger.

Ik heb ook tijd besteed aan het ontwikkelen van een visuele identiteit voor de site. Het logo dat ik creëerde bevat een kameleon met een triskelion-achtig symbool als staart:

- De kameleon symboliseert aanpassingsvermogen en veelzijdigheid
- De triskelion (drievoudige spiraal) staat voor groei, beweging en transformatie - kernwoorden in

Jos' werk

Op vrijdag valideerde ik of er een API beschikbaar was om posts te maken buiten de Ghost blog interface om. Deze functionaliteit bleek inderdaad aanwezig, wat de workflow voor content creatie zal verbeteren.

Testmigratie en afrondende werkzaamheden

De laatste opdracht van de week was het migreren van tests van MSTest naar xUnit voor het E-shop project. Dit proces werd bemoeilijkt door enkele ontbrekende projectreferenties en oudere .NET Framework projecten (<4.0). Na samenwerking met Dries om deze issues op te lossen, kon ik de tests succesvol migreren en valideren dat ze correct werken.

Tot slot heb ik nog wat Docker configuratie werk gedaan voor het StreamDesk project, waarmee ik de week productief afsloot.

Reflectie

Deze week heeft mij vooral geleerd hoe belangrijk het is om systematisch te debuggen. Door stap voor stap te werken en hypotheses te testen, kon ik complexe problemen oplossen die op het eerste gezicht ontmoedigend leken.

De projecten bij HolonCom bieden een uitstekende mix van technische uitdagingen die mijn vaardigheden verbreden - van plugin ontwikkeling en dockerisatie tot visuele identiteit creatie en testframeworks. Ik merk dat deze diversiteit mij dwingt om snel nieuwe concepten te leren en bestaande kennis in verschillende contexten toe te passen.

Comments

0 reactions



0 comments

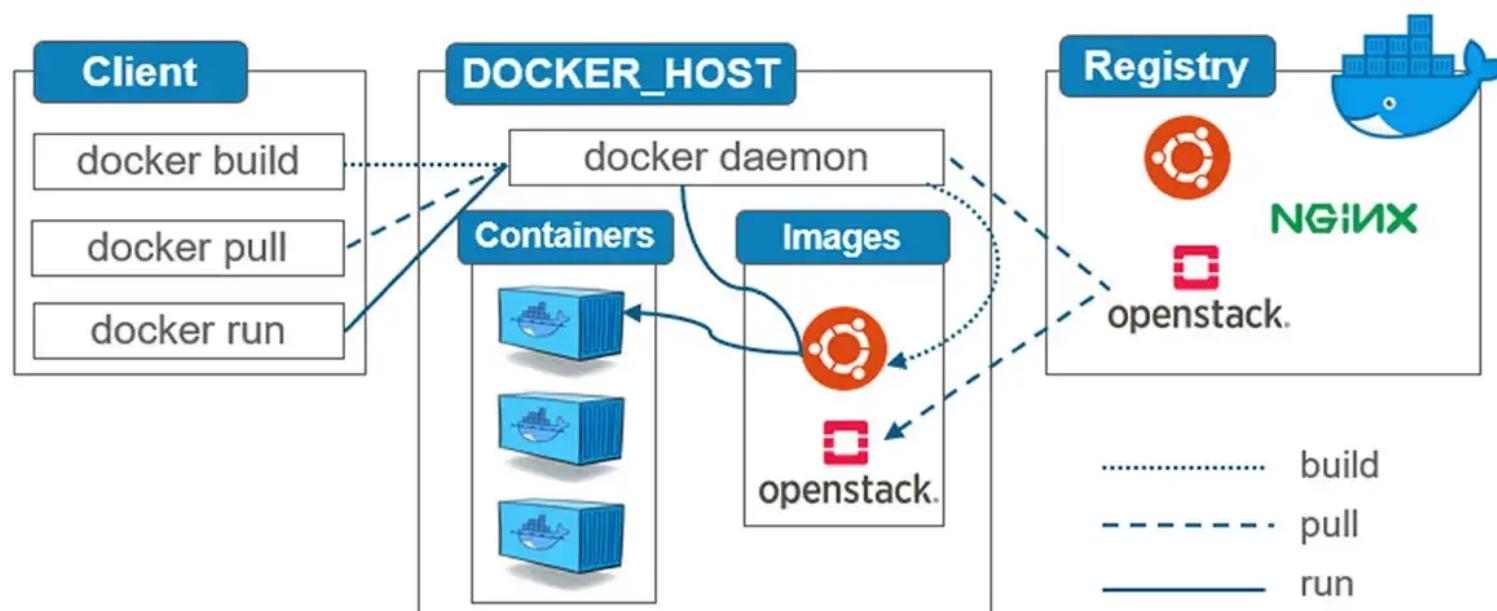
Write Preview Aa

Sign in to comment

M+

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[← Terug](#)

Week 12: Van NuGet Mysteries tot Docker Architectuur - Een Intensive Refactoring Week



Ian Mondelaers

May 22, 2025

Share:



Deze week stond volledig in het teken van diepgaande refactoring en infrastructuurverbeteringen. Van het oplossen van mysterieuze NuGet-problemen tot het opzetten van een complete Docker-architectuur - het was een week vol technische uitdagingen en leermomenten.

De Grote NuGet Mystery Ontrafeling

De week begon met het oplossen van een hardnekkig probleem dat me al een tijdje bezighield: bepaalde NuGet packages werden simpelweg niet gegenereerd. Na uitgebreid onderzoek ontdekte ik de oorzaak - een subtile maar cruciale fout in onze solution-structuur.

Het bleek dat we werkten met twee verschillende `.`sln`` files, waarbij de GitHub Action voor het builden en deployen van NuGet packages gebruikmaakte van een solution file die niet alle benodigde projecten bevatte. Dit was het resultaat van een eerdere refactoring.

De oplossing vereiste een grootschalige refactoring waarbij ik alle "NuGet-only" projecten moest toevoegen aan de juiste solution. Dit bracht echter nieuwe uitdagingen met zich mee, vooral rond interne projectreferenties die niet meer klopten door verschillend naamgeving tussen source- en NuGet-projecten.

Centraal Package Management met `Directory.Packages.props`

Een belangrijk onderdeel van deze week was het implementeren van centrale package versioning via een ``Directory.Packages.props`` file. Dit Microsoft-mechanisme zorgt ervoor dat alle NuGet package versies centraal beheerd worden, waardoor inconsistenties tussen projecten voorkomen worden.

Het refactoren van alle projectfiles om gebruik te maken van dit systeem was intensief werk - elke versie-specificatie moest uit de individuele projectfiles gehaald en naar het centrale bestand verplaatst worden. De voordelen zijn echter significant: consistente versioning, eenvoudiger onderhoud en verminderde kans op dependency conflicts.

Eerste Klantgesprek: Een Mijlpaal

Een persoonlijke mijlpaal deze week was mijn eerste echte klantgesprek met Jos over zijn blog website. Ondanks de initiële nervositeit verliep het gesprek uitstekend. Ik kon de volledige functionaliteit demonstreren, van het aanmaken van blogposts tot het beheren van tags en instellingen.

Een interessant verzoek kwam naar voren: seizoensgebonden banners voor de website. Jos wilde dezelfde afbeelding maar dan aangepast naar de kleurpaletten van verschillende sezoenen. Dit type functionele requirement gathering is precies wat je als software engineer regelmatig tegenkomt - apparentelijk eenvoudige verzoeken die eigenlijk veel zeggen over user experience en branding.

De feedback na het gesprek was waardevol: soms kwam ik wat te formeel over, waardoor er afstand ontstond. Een belangrijke les in stakeholder management - het vinden van de juiste balans tussen professionaliteit en toegankelijkheid.

Docker Architectuur: Van Monoliet naar Microservices

Het tweede grote project deze week was het opzetten van een complete Docker-architectuur voor onze StreamDesk applicatie. Wat begon als een enkele Dockerfile evolueerde naar een volledige multi-container setup met proper service discovery en netwerk isolatie.

De architectuur bestaat nu uit:

- **Gescheiden containers** voor server (API) en client
- **Service dependencies** via Docker Compose
- **Eigen netwerk** (streamdesk-net) voor geïsoleerde communicatie
- **Lokale services** voor RabbitMQ, MailHog, Seq en ArangoDB

Het implementeren van Docker networks was particularly interessant. Door een custom bridge network te definiëren, krijg je automatisch service discovery - containers kunnen elkaar vinden via hun servicenaam dankzij Docker's interne DNS. Dit elimineert hardcoded IP-adressen en maakt de setup veel flexibeler.

Technische Diepduik: Cherry Picking en Versioning

Een nieuw concept dat ik deze week leerde was **cherry picking commits**. Dit Git-mechanisme laat toe om specifieke commits van andere branches over te nemen zonder een volledige merge. Bijzonder handig wanneer je clean, atomic commits maakt - iets waar ik bewust op focus.

Voor NuGet versioning implementeerde ik een branch-gebaseerd systeem waarbij de pre-release identifier automatisch wordt aangepast op basis van de branch:

- `develop` branch → `1.0.0-develop.123`

- `release-candidate` branch → `1.0.0-rc.456`

Dit maakt het veel eenvoudiger om verschillende versies parallel te ontwikkelen en te testen.

Uitdagingen en Leermomenten

De grootste uitdaging deze week was het debuggen van de Docker setup. Van DNS-problemen door VPN-configuraties tot missing configuration files - elk probleem leerde me meer over containerization en Linux-omgevingen.

Een cruciale les was het besef dat Docker containers volledig self-contained moeten zijn. Afhankelijkheden van externe services (zoals RabbitMQ op specifieke IP-adressen) maken je setup fragiel. Door alles lokaal op te zetten via Docker Compose werd de architectuur veel robuuster.

Reflectie en Vooruitblik

Deze week toonde duidelijk aan hoe verschillende aspecten van software development met elkaar verbonden zijn. Een simpel NuGet-probleem leidde tot solution refactoring, wat weer package management verbeteringen triggerde, wat uiteindelijk resulterde in betere versioning practices.

Het klantgesprek was persoonlijk een grote stap - van puur technisch werk naar stakeholder

communicatie. De feedback helpt me groeien in deze soft skills die net zo belangrijk zijn als technische vaardigheden.

Voor volgende week staat het finaliseren van de Docker setup op de planning Het gevoel dat je als software engineer constant leert en groeit, dat vind ik het mooiste aan dit vak.

Comments

0 reactions



0 comments

Write Preview Aa

Sign in to comment

M

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#) [Blog](#) [Over Mij](#) [Contact](#) [Log In](#) [Terug](#)

[Klantgesprek](#)[ProjectConfiguration](#)[Docker](#)

Week 13: Docker Mastery, Client Evolution en daadkracht - Van SSL Struggles tot Porto

**Ian Mondelaers**

May 22, 2025

Share:



Week 13 werd gekenmerkt door de voltooiing van onze Docker-architectuur en een interessante pivot in een klantproject. Van het oplossen van complexe SSL-configuraties tot het transformeren van een blog naar een persoonlijk journal - deze week toonde aan hoe veelzijdig software development kan zijn.

Docker SSL Debugging: Een Technische Deep Dive

De week begon met het aanpakken van persistente SSL-problemen in onze Docker setup. Het probleem was subtiel maar kritiek: Kestrel, de .NET webserver, probeerde automatisch HTTPS-endpoints op te zetten, wat tot certificaat-gerelateerde crashes leidde in een containeromgeving.

De oplossing vereiste een gerichte Kestrel-configuratie waarbij ik expliciet instructies gaf om alleen op poort 80 te luisteren voor HTTP-verkeer. Dit type low-level webserver configuratie is cruciaal in

containerized environments waar je volledige controle wilt over netwerkverkeer en certificaatbeheer.

Een leerzame fout die ik tegenkwam was een Docker container naming conflict. Toen ik een service hernoemde, maar de `**container_name**` hetzelfde liet, probeerde Docker een nieuwe container aan te maken met een naam die al bestond. Dit illustreert perfect waarom consistency in naming conventions zo belangrijk is in infrastructure-as-code.

Van Blog naar Journal: Requirements Evolution

Een fascinerende ontwikkeling deze week was de pivot van Jos' blog naar een journal-format. Dit is een perfect voorbeeld van hoe klantbehoeften evolueren tijdens development. Waar oorspronkelijk een traditionele blog-layout werd gevraagd, bleek een meer persoonlijke journal-structuur beter te passen bij de use case.

De nieuwe requirements waren specifiek:

- **Centrale datum en titel** als primary focus
- **Tag-gebaseerde navigatie** rechts in de sidebar
- **Filtering functionaliteit** waarbij klikken op een tag alle gerelateerde posts toont

Het interessante hier was dat ik te maken kreeg met een Ghost CMS compatibility issue - een theme

update had sommige templates gebroken. Dit leidde tot een Ghost update en het zoeken naar een geschikt journal theme. Een goede reminder dat dependency management zich uitstrekkt tot CMS platforms en themes.

Containerization Strategy: Source vs NuGet

Een significante architecturale beslissing deze week betrof onze package distribution strategy. Na uitvoerig overleg met mijn mentor werd besloten om af te stappen van een NuGet-based approach in Docker containers ten gunste van een source-code gebaseerde build.

Dit had verschillende technische implicaties:

- **Transitieve dependencies** werden eenvoudiger te beheren
- **Version conflicts** kwamen minder voor
- **Build reproducibility** verbeterde significant
- **Container size** nam wel toe door de volledige source tree

Het refactoring proces was intensief - via Beyond Compare 5 moest ik zorgvuldig bepalen welke projecten daadwerkelijk nodig waren voor de solution. Elke build cycle onthulde weer nieuwe missing dependencies, wat een iteratief debug proces vereiste.

Command Worker Containerization

Het tweede grote Docker project was de Command Worker service. Dit was uitdagender dan verwacht door verschillende target framework incompatibiliteiten tussen NuGet packages en lokale projecten.

Een cruciaal inzicht was dat andere developers deze problemen niet ondervonden omdat zij via source builds werkten, waarbij de externe source code mee in de solution zat. NuGet packages daarentegen dwingen strikte version compatibility af, wat subtiele problemen kan blootleggen die anders onopgemerkt blijven.

Het MSBuild `SolutionDir` variabele probleem kwam opnieuw naar boven, wat ik oploste door de waarde hard-coded mee te geven in de publish step. Niet de elegantste oplossing, maar wel pragmatisch en effectief.

Nginx vs Python Static File Serving

Een interessante architecturale discussie ontstond rond static file serving voor onze Blazor client. Mijn eerste implementatie gebruikte Python als workaround, maar dit was suboptimaal voor performance en best practices.

De terugkeer naar Nginx als reverse proxy was de juiste keuze.

Nginx is specifiek ontworpen voor dit type workload en biedt:

- **Superior performance** voor static content
- **Built-in caching** mechanisms
- **Proper HTTP headers** voor browser caching
- **Production-ready** configuratie opties

Technical Learnings: Docker Volumes en Mounting

Een praktisch inzicht deze week was het verschil tussen Docker volumes en bind mounts. Voor de ArangoDB configuratie switchte ik van named volumes naar bind mounts, wat directe toegang geeft tot database files op de host machine.

Het mounting principe in Docker - van host naar container - lijkt simpel maar heeft belangrijke implicaties voor data persistence en debugging. Zonder een proper `docker cp` strategy kan data extraction complex worden.

Beyond Technical: Daadkracht en Persoonlijke Groei

Een memorabel moment deze week was het gesprek met mijn mentor over "daadkracht" - de kunst van plannen omzetten in actie. Dit kwam naar boven toen ik mijn interesse uitsprak in lange wandeltochten zoals de Camino de Santiago.

Zijn advies was direct en waardevol: "Doe het, laat het niet bij woorden." Deze mentaliteit - van planning naar execution - is even relevant in software development als in persoonlijke groei.

Project Completion: Attention to Detail

Het afronden van het Docker project vereiste verschillende kleine maar belangrijke refinements:

- **Service naming** consistency across alle componenten
- **Mount configurations** optimization voor verschillende services
- **Folder structures** die logisch en maintainable zijn
- **Named volumes** conversie naar bind mounts waar appropriate

Deze "finishing touches" lijken misschien trivial, maar ze maken het verschil tussen een werkende proof-of-concept en een production-ready solution.

Reflectie: Van Problem Solving naar Solution Architecture

Deze week illustreerde mooi de evolutie van junior naar medior developer skills/mindset. Waar ik eerder focuste op individuele problemen oplossen, ben ik nu meer bezig met system-wide implications en architectural decisions.

Het SSL debugging was niet alleen een technical fix, maar leidde tot een beter begrip van container networking. De Ghost CMS update was niet alleen een compatibility fix, maar een les in dependency management. De Docker source code strategy was niet alleen een build optimization, maar een fundamental architectural decision.

Deze holistische manier van denken - waar elk technical decision wordt bezien in de context van het gehele systeem - is precies wat ervaren software engineers onderscheidt van beginners.

Comments

0 reactions



0 comments

Write

Preview

Aa

Sign in to comment



Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#) [Blog](#) [Over Mij](#) [Contact](#) [Log In](#) [Terug](#)

[SoftwareEngineering](#)[Docker](#)[Deployment](#)

Week 14: Deployment pipelines, Docker integratie en het afronden van losse eindjes

**Ian Mondelaers**

May 22, 2025

Share:



Deze week stond volledig in het teken van consolidatie en optimalisatie. Met het einde van mijn stage in zicht, richtte ik me op het afwerken van openstaande taken en het implementeren van cruciale deployment-oplossingen. Deze week bracht interessante technische uitdagingen met zich mee, van cross-platform compatibiliteit tot automated versioning systemen.

VS Code Extension deployment en Docker voorbereidingen

De week begon met een grondige evaluatie van mijn takenlijst - een essentiële stap om ervoor te zorgen dat ik geen losse eindjes achterlaat. Een van de eerste taken was het "vernederlandsen" van Jos' blog website, waarbij ik het Engels thema moest lokaliseren naar het Nederlands.

Het werken met remote development via Webmin gaf me waardevolle inzichten in serverconfiguratie. Door een poortverbinding op te zetten kon ik direct vanuit VS Code wijzigingen aanbrengen in de gehoste bestanden. Dit workflow patroon - waarbij lokale development tools worden gekoppeld aan remote servers - is een krachtige methode die ik vaker zal toepassen.

Een belangrijke mijlpaal was het succesvol deployen van mijn VS Code plugin naar de Visual Studio Marketplace. Het VSCE (Visual Studio Code Extension) proces leerde me veel over software distributie:

- **VSCE tooling:** De command-line tool voor packaging en publishing van extensies
- **VSIX packaging:** Het creëren van distribueerbare pakketbestanden
- **Marketplace deployment:** Het volledige proces van submission tot publicatie

Parallel hieraan werkte ik aan het oplossen van build issues in het HolonCom.Core.All project. Deze debugging sessie bracht twee veel voorkomende problemen aan het licht:

1. **Verkeerde variabel referenties:** Een klassiek voorbeeld van waarom code review cruciaal is
2. **Nuget versioning conflicts:** Oude versies die niet de nieuwste codebase reflecteerden

Nuget versioning complexiteit en Docker implementatie

De dag begon met een inspirerend gesprek over de Camino de Santiago, wat me deed realiseren hoe waardevol de informele momenten met collega's zijn voor team building en kennisdeling.

Het technische hoogtepunt was het tackelen van een complex versioning probleem. De non-semantische versioning "3.0.1.5-develop.0.89" bracht interessante uitdagingen met zich mee. Ik leerde over version ranges in NuGet:

```
[3.0.1-3.0.2) // 3.0.1 inclusief, 3.0.2 exclusief
```

Deze notatie, hoewel op het eerste zicht verwarrend, biedt krachtige mogelijkheden voor dependency management.

Voor de Docker implementatie in de command worker hanteerde ik de "fail fast" methodologie. Deze aanpak - waarbij je bewust alle externe referenties verwijdert en systematisch de benodigde dependencies toevoegt - zorgt voor een cleane, minimale codebase. Het resultaat was een geoptimaliseerde solution met alleen de essentiële componenten.

Een cruciale les betrof **case sensitivity verschillen** tussen operating systems:

- **Windows:** Case-insensitive voor bestandspaden
- **Ubuntu/Linux:** Case-sensitive voor bestandspaden

Dit verklaarde waarom builds lokaal slaagden maar faalden in de CI/CD pipeline. De normalisatie van folder- en projectnamen was essentieel voor cross-platform compatibility.

Git configuratie en source code refactoring

Merge conflicts brachten interessante Git gedragingen aan het licht. Het commando `git config core.ignoreCase false` was een game-changer voor cross-platform development. Deze configuratie maakt Git case-sensitive op Windows, wat consistentie waarborgt met Linux-gebaseerde CI/CD omgevingen.

Een fascinerende Git-eigenschap die ik ontdekte: **Git slaat alleen bestanden op, geen folders.** Folders worden behandeld als namespaces. Deze architecturale beslissing verklaart waarom het hernoemen van folders in Git complex kan zijn.

Het refactoren van de E-shop solution vereiste uitgebreide projectfile synchronisatie. Projecten die lang alleen als NuGet packages waren gebruikt, hadden verouderde source code projectfiles. Deze asymmetrie tussen NuGet en source-only projectfiles is een veelvoorkomend probleem in grote codebases.

Visual Studio vs Command Line building

Een belangrijke ontdekking betrof de verschillen tussen Visual Studio en command-line builds. Visual Studio bevat standaard target files die niet op het bestandssysteem aanwezig zijn, maar wel beschikbaar zijn tijdens compilation. Dit kan leiden tot:

- **Compatibiliteitsproblemen** met oudere projectfiles
- **Inconsistente build gedragingen** tussen development en CI/CD omgevingen
- **Missing dependency issues** bij manual compilation

PowerToys' PowerRename tool bewees zijn waarde voor bulk refactoring operaties. Dit soort tools zijn onmisbaar voor grote codebase transformaties.

Automated versioning en multi-targeting challenges

De implementatie van automated build versioning voor Core.Arango was een complexe maar waardevolle oefening. Het systeem werkt als volgt:

- **Version prefix:** 3.0.1 (manueel beheerd)
- **Build number:** Auto-increment bij elke push
- **CI/CD integration:** Automatische versie updates

Kritische overwegingen voor dit systeem:

1. **Infinite loop prevention:** `'[skip ci]'` commits voorkomen recursieve triggers
2. **Timing coordination:** Versie updates voor NuGet packaging

De uitbreiding van mijn VS Code plugin met flexible file detection binnen KnowledgeGraph folders vereiste geavanceerde bestandssysteem navigatie. Deze feature verhoogt de gebruiksvriendelijkheid aanzienlijk.

Ghost project lokalisatie bracht me in contact met moderne i18n (internationalization) patronen:

- **JSON-based translations:** Gestructureerde key-value pairs
- **Template integration:** `{{t "key"}}` helpers voor dynamic content
- **Fallback mechanisms:** en.json als backup voor nl-be.json

Het werken met multi-target frameworks in .NET bracht een belangrijk inzicht: ambiguïteit in publish commando's kan builds doen falen. De oplossing - explicit framework specification met `'-f net9.0'` - is een best practice voor CI/CD pipelines.

Reflectie en geleerde lessen

Deze week onderstreepte het belang van:

1. **Cross-platform awareness:** OS-specifieke gedragingen kunnen subtile maar kritieke problemen veroorzaken
2. **Automated tooling:** Deployment pipelines en versioning systemen verminderen menselijke fouten
3. **Clean architecture:** "Fail fast" methodologieën leiden tot optimale codebases
4. **Documentation:** README files en proces documentatie zijn cruciaal voor knowledge transfer

De balans tussen technische diepgang en praktische implementatie was deze week perfect. Elke uitdaging bracht waardevolle inzichten die direct toepasbaar zijn in toekomstige projecten.

Comments

0 reactions



0 comments

Write

Preview

Aa

Sign in to comment

M▼

Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.

[Home](#)[Blog](#)[Over Mij](#)[Contact](#)[Log In](#) [Terug](#)

[SoftwareEngineering](#)[Deployment](#)[InternshipConclusion](#)

Week 15: De laatste loodjes - FTP deployment, synchronisatie en een afscheid

**Ian Mondelaers**

May 23, 2025

Share:



Deze markeerde het einde van een transformerende reis als software engineering stagiair bij HolonCom. Deze laatste week was een intense mix van technische consolidatie, deployment optimalisaties en persoonlijke reflectie. Van FTP protocol implementatie tot de emotionele impact van een afscheid - deze week had het allemaal.

Nuget cleanup en FTP verkenning

De week begon met waardevolle career guidance van Anna over CV optimalisatie en LinkedIn professionalisering. Deze soft skills zijn net zo belangrijk als technische competenties - een les die vaak onderbelicht blijft in academische curricula.

Technisch gezien focuste ik op **automated cleanup van NuGet packages** op Gitea. Door regex patterns te verfijnen, kon ik een robuuster systeem implementeren dat toekomstbestendig is. Dit

type maintenance automation is cruciaal voor lange-termijn codebase gezondheid.

Een interessante uitdaging was de **assembly versioning** van de E-shop. Het probleem: dynamische assembly files met timestamps die constant Git commits vervuilen. Hoewel MinVer een logische oplossing leek, veroorzaakte het onverwachte build errors - waarschijnlijk caching-gerelateerd. Dit illustreert een belangrijke ontwikkelaarswaarheid: soms is de elegante oplossing niet altijd de juiste oplossing op dat moment.

Mijn eerste kennismaking met **File Transfer Protocol (FTP)** was fascinerend. FTP, een TCP/IP-gebaseerd netwerkprotocol voor bestandsoverdracht, opende nieuwe deployment mogelijkheden. De architectuur vereist drie essentiële componenten:

- **Host:** Server endpoint
- **User credentials:** Authenticatie
- **Password:** Secure access

Solution normalisatie en FTP implementatie

De **systematische normalisatie van Solution (.sln) files** was een intensieve maar waardevolle oefening. Voor elk project (Core.All, Command Worker, E-shop) creëerde ik drie distinct solutions:

1. **Source-only:** Pure source code references
2. **NuGet-only:** Package-based dependencies
3. **Hybrid:** Source code + external library source

Deze structuur, met alfabetische project ordering, maakt cross-solution vergelijking met tools zoals Beyond Compare 5 trivial. Het is een perfect voorbeeld van hoe systematische organisatie toekomstige maintenance drastisch vereenvoudigt.

De **FTP action implementatie** bracht interessante deployment inzichten:

Override mechanisme: De action vergelijkt lokale bestanden met `FTP-deploy-sync-state.json` metadata. Wijzigingen in inhoud, grootte of modificatiedatum triggeren automatische uploads met bestandsoverschrijving.

Directory management: De `server-dir` parameter maakt automatische submappen creatie mogelijk, wat flexible deployment structuren ondersteunt.

Synchronisatie uitdagingen en persoonlijke groei

Mijn voorlaatste dag bracht complexe **cross-project synchronisatie** uitdagingen. Het synchroniseren van Streamdesk en Command Worker code onthulde een rookie mistake: simpel kopiëren van

bestanden houdt geen rekening met verwijderde files aan de source kant.

Deze fout resulteerde in:

- **Duplicate definitions:** Conflicterende code definities
- **Broken legacy code:** Outdated references
- **Superfluous dependencies:** Overbodige project references

Beyond Compare 5 werd mijn redding voor deze complexe refactoring. De tool maakte systematische vergelijking en synchronisatie mogelijk.

Docker-specifieke uitdagingen ontstonden door MSBuild variabelen zoals `\$(SolutionDir)` die niet resolvable zijn in containerized environments. Ook server naming differences tussen projecten vereisten careful configuration management.

De implementatie van een **gestandaardiseerde folder structuur** volgde best practices:

```
[project-name]/  
|   └── source/  
|       └── tests/
```

Deze organisatie verbetert code navigatie en onderhoudbaarheid aanzienlijk.

deployment, reflectie en afscheid

Mijn laatste dag bij HolonCom begon met het oplossen van **NuGet-only solution build errors**. De root cause: onvolledige refactoring tijdens Core.All integratie. Deze ervaring onderstreepte het belang van thorough testing na structural changes.

FTP deployment actions voor Command Worker server/client en E-shop componenten vereisten careful credential management. Initial testing met test accounts onthulde MSBuild variable resolution issues in CI/CD pipelines.

De oplossing voor `\$(SolutionDir)` resolution:

```
- name: Publish  
  run: dotnet publish -p:SolutionDir="${{github.workspace}}/"
```

External library referencing in publish steps vereiste expliciete configuration:

```
- name: Configure NuGet source  
  run: dotnet nuget add source [HolonCom-source]
```

Additional technical accomplishments:

- ArangoDB image backup naar Gitea registry voor disaster recovery
- SMTP server configuratie voor Ghost project email functionality
- Config externalization door Docker volume mounting

Persoonlijke transformatie en dankbaarheid

Reflecterend op mijn eerste dag versus laatste dag, de transformatie is aanzienlijk. Ik kwam binnen als een nerveuze student en vertrek met meer zelfzekerheid, groei op persoonlijk en professioneel gebied én met meer vakjes.

Dankwoord

Aan het geweldige team van HolonCom - jullie hebben meer dan een stageplaats geboden. Jullie

hebben een leeromgeving gecreëerd waarin ik kon experimenteren, falen, leren, groeien en (persoonlijke) grenzen kon verleggen.

Robrecht, als mentor heb je me niet alleen technische skills bijgebracht, maar ook belangrijke levenslessen en inzichten. Je manier om groei op alle vlakken te faciliteren, zelfs in uitdagende momenten, is een gift. Dank je voor je geduld, expertise en voor het pushen van mijn grenzen.

Anna, je warmte en wijsheid hebben deze ervaring extra speciaal gemaakt. Je career advice en oprechte bezorgdheid voor mijn groei als persoon waren onschatbaar. Ik kan niet wachten om je te vertellen over mijn Camino.

Dries, dank je voor je technische ondersteuning en inzichten. Working alongside experienced developers like you made me a better programmer.

Deze stage bij HolonCom was transformationeel. De skills, zelfzekerheid en het netwerk dat ik hier heb opgebouwd, zullen de basis vormen voor mijn verdere carrière.

Met oprechte dankbaarheid en nieuwsgierigheid voor de toekomst,

"Not all those who wander are lost" - en soms leiden wandelingen naar de mooiste bestemmingen.

Comments

0 reactions



0 comments

Write

Preview

Aa

Sign in to comment



Sign in with GitHub

© 2025 DenGian Blog. All rights reserved.