

Top 10 erfaringer med opsætning af Azure DevOps Testmodul

Jeg vil lige starte med en lille disclaimer – dette er ikke det universelle svar på hvordan man organiserer test i Azure DevOps 😊, men det er lidt praktiske erfaringer jeg har gjort mig her på det sidste. Samtidig skal det nævnes at jeg i det følgende kun fokuserer på den manuelle test da det var opgaven, og at jeg kun er almindelig bruger – jeg har ikke haft administratorrettigheder i denne proces.

Det sidste stykke tid har jeg nemlig brugt lidt kræfter på at forstå Azure DevOps testplan modulet, og det har været et rigtig interessant forløb hvor jeg blandt andet er kommet til konklusionen om at spejderbevægelsens motto "learning by doing" giver så meget mening 😊.

Tip nr. 1 Få styr på hvad du egentlig har brug for

Man kan hurtigt fare vild og drukne i ustruktureret konfiguration af Azure hvis man ikke har bare et minimum af en plan, også her er det smart at starte med "why"! Hvorfor er det vi vil have test ind, hvad er det for en værdi det skal skabe for os og for teamet at test nu bliver integreret med de andre aktiviteter i Azure DevOps?

Inden man for alvor går i gang, så er det altså om at skabe et overblik over hvad det egentlig var for nogen krav jeg har – bare sådan på overskrift basis, f.eks.:

- Test og agile skulle være tæt forbundne – kontinuerlig fokus på det enkelte sprint
- Testaktiviteter synlige på boardet, ikke bare som en task men så man kan se afviklingsstatus
- Etablering af manuel regressionstest suite (mit testbibliotek)
- Genbrug af testcases hvor relevant
- Sporbarhed mellem story-test-defect
- Transparens i forhold til status og fremdrift
- Mulighed for at kunne eksportere testcases til Excel eller lignende
- Brainstorming på test når man starter på en story
- ...

Tip nr. 2 Lær og Eksperimenter

Jeg startede op på ganske traditionel vis; læste noget dokumentation hos Microsoft, så en flok tutorials på youtube... men jeg havde stadig flere spørgsmål end svar:

- Hvordan får vi skabt et fokus der ikke kun er på sprints, men også på at skabe en regressions test suite – også en manuel en af slagsen?
- Hvordan får jeg skabt et godt sammenhæng mellem det der sker i den "agile del" af Azure og det der sker i testplan delen
- Hvordan får jeg max transparens så ikke kun tester og testmanager har styr på test?
- Hvordan er sammenhængen mellem story, defect, og test case i Azure i praksis?
- Hvad nu når jeg har en formel release efter et antal sprints hvor features skal testes i helhed og der skal laves regressionstest?
- Hvad gør jeg med User Accept Test – ud mod slutbrugerne der ikke har adgang til Azure DevOps
- Og mange flere....

Men så er der jo kun en ting at gøre; At lære ved at eksperimentere i små iterativer virker også her. For hvert krav jeg havde satte jeg mig til at "lege" med systemet, prøvede forskellige løsninger af – jeps jeg testede testmanagement modulet.

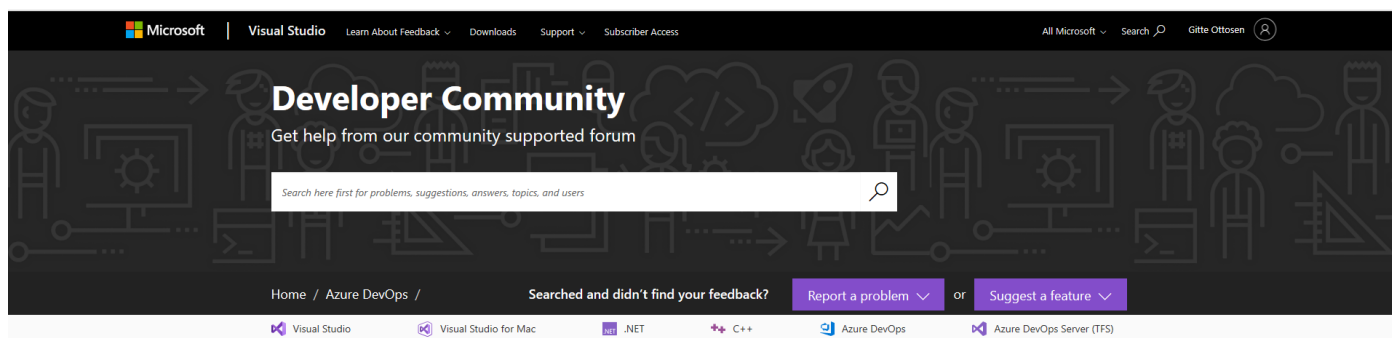
Jeg synes egentlig det er lykket mig at lave en ret god opsætning nu, men en klar anbefaling er: lav en sandkasse hvor du kan lege uden at påvirke dem der sidder og arbejder derinde i "rigtige projekter".

Tip nr. 3 Brug Community

En af de ting jeg har haft virkelig stor gavn af, er Microsofts community for Azure DevOps. Det er helt enormt den hjælpsomhed jeg har oplevet derude, både fra Microsoft supportere og andre brugere. Dels når jeg har haft problemer med at finde funktionalitet eller forstå det jeg har fundet, og dels når jeg gerne ville aflevere fejl/mangler eller feature anmodninger.

En ting er, at de svarer lynhurtigt, der er svar samme dag hver gang, men de er faktisk også rigtig lydhøre for at få ideer til nye features hvis man har gode ideer. Problemet jeg fandt nede ved tip nr. 6 er f.eks. blevet godkendt som en feature, mere om dette senere.

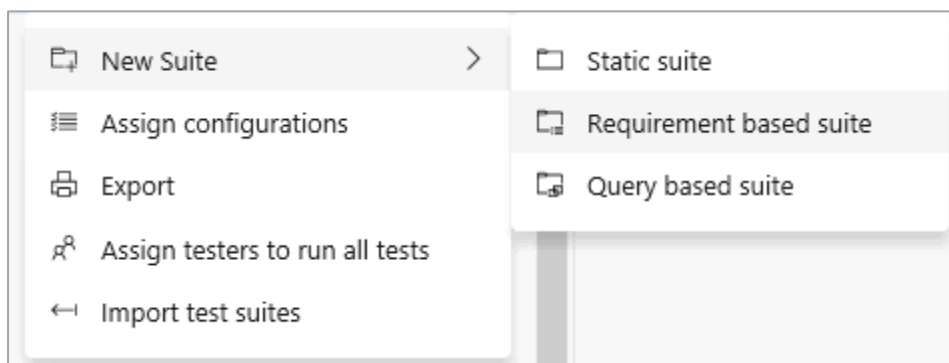
Så en klar anbefaling må være; opret dig som bruger og vær aktiv i deres community.



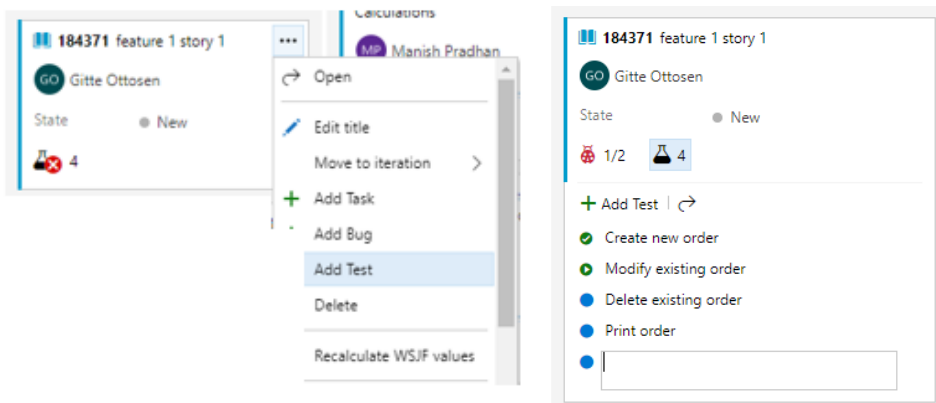
Tip nr. 4 Få styr på sporbarhed mellem stories og testafvikling

Her er der faktisk flere smarte features at tage fat i;

Requirement based test suites i testplanen. Lav en testplan pr sprint og brug funktionen til at skabe en "folderstruktur" med en folder pr user story. Skaber sporbarhed mellem story og test, hver gang du opretter en testcase i en folder, så bliver den automatisk oprettet som "related work" i user storien.



Brug "add test" i "..." menuen for en user-story i Board. Lav one-liners når du brainstormer over hvad du skal have testet for en story, f.eks når du taler med forretningen eller diskuterer med en udvikler om løsningen. Hver one-liner er en testcase i din testplan, automatisk oprettet ovre i den rette folder.



Tip nr 5 forstå forskellen på copy og clone

En helt grundlæggende funktionalitet for mig er at hvis jeg vil genbruge en testcase, f.eks hvis noget lignende funktionalitet også bliver udviklet i et andet modul af systemet, så skal jeg kunne kopiere testcasen så jeg ikke retter i den oprindelige testcase når jeg begynder at redigere den "nye" testcase. Her er det virkelig vigtigt at forstå at når der f.eks. er en funktion der kan importere en eksisterende suite, eller "add existing test cases" knappen så er det *Referencer* til den oprindelige suite eller testcase der bliver skabt, det er ikke en ny kopi du bare kan arbejde videre i.

For at skabe en egentlig kopi af testcasen skal du åbne din testcase, klikke på de tre prikker i øverste højre hjørne og vælge "create copy of work item". Det er en reel kopi og ikke en reference du her får lavet.

Så beslut dig til hvad du vil – en hel frisk testcase med input fra en gammel, eller bare en reference til den så rettelser bliver nedarvet mellem de enkelte test suites.

På grund af det med referencerne så betyder det at du vælger "remove" så heller ikke at du sletter testcasen som sådan, kun referencen til den. Den oprindelige testcase ligger stadig som et work item. Hvis du vil slette en testcase HELT så skal du have specifik permission til det – der er en permission der hedder "permanently delete work items".

Tip nr 6 Skab transparens på status på storyniveau

Pas på du ikke får sat en løsning op hvor kun testeren eller testmanageren ser noget til teststatus og fremdrift, vores arbejde er ikke en separat proces – det er aktiviteter i en fælles agil proces. Her kan man gøre flere ting;

Status på test synlig for den enkelte story. Her var jeg udfordret af at layout for story i board vs sprint board er ikke det samme... måske specifikt hos min kunde jeg er ikke sikker. Men i Board er der faktisk et ikon på storyen der klart viser hvad test status på afvikling er, det har jeg ikke i sprintboardet.



Ovenstående skyldes at vi her taler om to forskellige ting; **Test state vs Test outcome**. Hvor test state er design/ready/closed for testcasen – altså i forhold til at få den udarbejdet, så er Test outcome som du ser på det højre screenshot fra forsiden af en user story "outcome" af test afviklingen, altså passed, failed, other Hvis du åbner

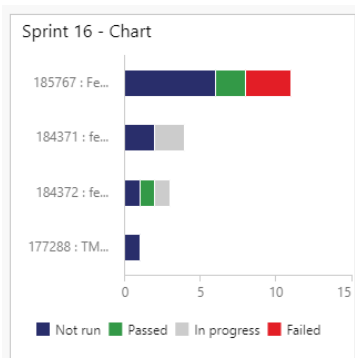
historien og kigger på "related work" så vil du også se at det er test state man bruger. Jeg tillod mig at rejse en feature request til Microsoft om at det burde være outcome og ikke state 😊

Tip nr 7 Status på test i sprintet, eller releasen

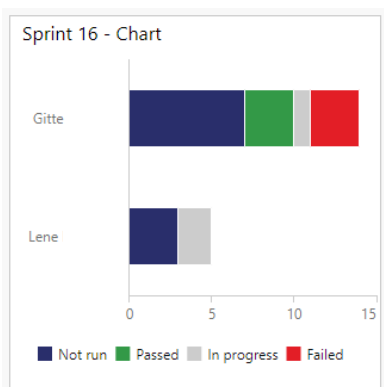
I tip 6 var fokus på den enkelte story, men du har jo også brug for at have overblik på status for hele sprintet eller måske for en egentlig testfase. Også her er der en nem måde at få skabt noget transparens;

1. Marker selve rodfolderen i din testplan
2. Vælg Chart tab'en
3. Her kan du nu konfigurere grafer efter dit eget hoved

Jeg har for eksempel lavet en oversigt over status pr user story

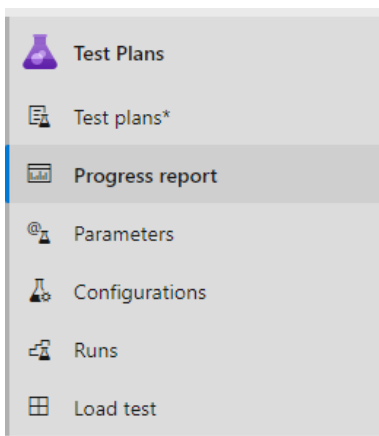


Det kunne også være en status pr tester (det er jeg dog personligt ikke vild med 😊)



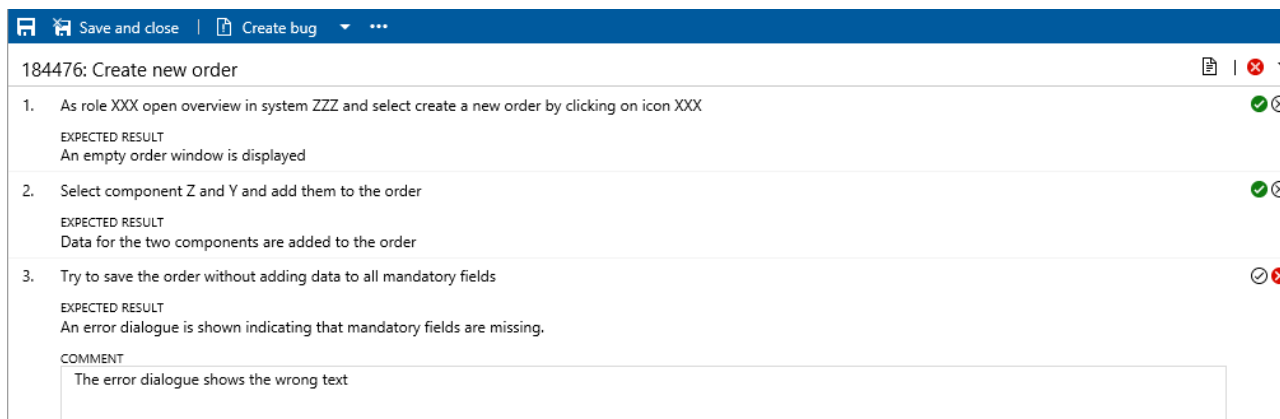
Men det er op til dig; hvad er dit behov – eller måske rettere hvad er dine interessenters behov for status.

Men der er selvfølgelig også de to områder i testmodulet der hedder "Progress report" og "Runs":

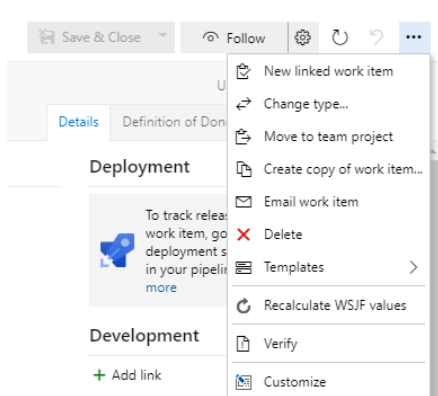


Tip nr 8 skab sporbarhed mellem stories, tests og fejl

Opret fejl fra test execution og skab din sporbarhed mellem testcase(og dermed user-story) og fejl. Og få "steps to reproduce" gratis foræret samtidig med at du kan tage screendumps, lave videoer mv derinde fra. Bemærk dog at den del med screendumps og videoer afhænger af de add-ons du har tilføjet, se tip nr 10.



Verificer den rettede fejl inde fra fejlen! Vælg "verify" fra "... " i øverste højre hjørne og så starter testcasen helt automatisk op i runner, og status opdateres i testplan når du er færdig. Ifølge den dialog der kommer, så burde den også lukke fejlen... det skete dog ikke de gange jeg har testet det.



Tip nr 9 Let søgningerne – brug også tags i test cases

Brug tags også i dine test cases. Det gør det langt nemmere at fremsøge testcases til genbrug, f.eks hvis du skal lave regressionstest på specifik funktionalitet. Husk i den forbindelse **at du kan editere flere testcases på en gang!** Det er nu meget handy når f.eks der skal sættes tags på en hel række testcases på bagkant 😊

Tip nr 10 Vær opmærksom på add-ons

Der er MANGE add-ons til Azure DevOps, og jeg vil slet ikke påstå at jeg har helt styr på dem alle. Men jeg har fundet et par stykker frem som dels hjælper rigtig meget i det daglige arbejde, og dels giver mig mulighed for at opfylde et krav jeg havde i min nuværende kontekst.

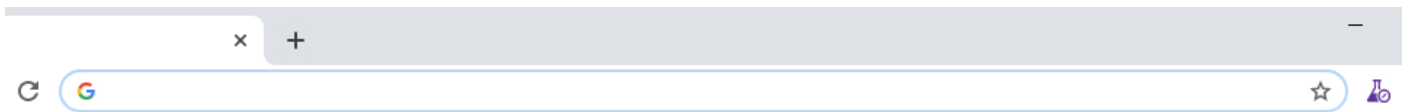
Screenshots og feedback

Hvis du tester applikationer der er web baserede så er der en rigtig fin add-on der hedder Test & Feedback. Den giver dig mulighed for at samle informationer op når du afvikler test og finder fejl/observationer du skal have dokumenteret. For øjeblikket er den kun til Chrome og Firefox, men er under udvikling til Edge også. Jeg har installeret den til min Chrome browser og dels er det vældig nemt at få sat op (hver enkelt tester skal gøre det, det er ikke på server siden).

Gå til <https://marketplace.visualstudio.com/items?itemName=ms.vss-exploratorytesting-web>

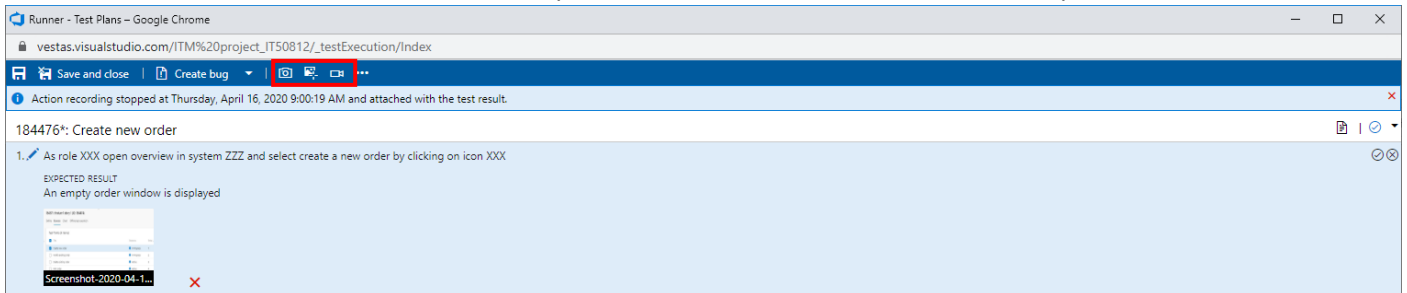
Her finder du den extension du skal downloade – husk at gøre det fra en Chrome eller Firefox browser 😊

Når du har installeret den så har du fået tilføjet et ikon til højre for din URL i browseren:



Nu skal du have forbundet den til din Azure DevOps instans så klik på den og vælg Settings. Skriv URL til din instans af Azure, og derefter får du mulighed for at vælge det projekt du er tilknyttet.

Når du herefter afvikler en testcase fra en testplan med din testrunner, så har du fået et par ekstra ikoner:



Der er tre forskellige ikoner:

1. Screenshot
2. Action log
3. Video

Og når disse benyttes, så bliver dit screenshot, din log eller din video tilknyttet testcasen. I screenshottet ovenfor kan du se at jeg har haft valgt et trin i testcasen da jeg trykkede på screenshot, så det jeg har gemt bliver tilknyttet det specifikke trin. Hvis du efterfølgende opretter en bug med knappen "Create Bug" så vil disse informationer komme med over i fejlbeskrivelsen og dermed sikre en bedre kvalitet af den fejl du registrerer.

Eksport af testcases

Jeg har et specifikt krav i min nuværende kontekst om at jeg skal kunne trække testcases UD af Azure igen, dels fordi de slutbrugere vi har ikke har direkte adgang til Azure DevOps og dels fordi det skal overdrages til Operations når vi lukker projektet og skal fastholdes i et andet system (Solution Manager). Det krav har givet mig lidt udfordringer, men igen kom det fantastiske Microsoft Community til med gode råd. Der er nemlig en add-on der hedder "Offline Test Execution" som kan hjælpe. Denne kræver så at du får installeret den på server siden, altså at du har administratorrettigheder – men jeg havde heldigvis en ven der kunne 😊 og hvad giver den så? Jo når du åbner en testsuite så har du nu fået en ekstra tab udover design, execute og chart. Nu er der også en der hedder Offline Test execution.

184371 : feature 1 story 1 (ID: 184474)

Define Execute Chart **Offline test execution**

Export ▾

Filter export to:

Include Selected suite and child suites ▾

Config -- All -- ▾

Tester -- All -- ▾

Export to Excel

Test Case	Title	Test Point	Configuration
184476	Create new order	29266	Windows 10
184477	Modify existing order	29267	Windows 10
184478	Delete existing order	29268	Windows 10
184586	Print order	29269	Windows 10

Her har du mulighed for at sætte et filter op med hvad du vil have eksporteret, og dette ender så i en Excel fil du kan bruge til f.eks UAT Hvis det skal være udenfor Azure.

Den vågne læser bemærker nok at der hvor der står Export er en lille pil, altså en dropdown. Det er fordi det jo er lavet til at man kan afvikle offline... men resultatet skal jo tilbage til Azure. Så du kan såmænd også importere igen når testen af afviklet så du får testresultatet ind i din plan.

184371 : feature 1 story 1 (ID: 184474) [Help](#)

Define Execute Chart **Offline test execution** ↗

Import ▾

1. Upload your Excel File to import the test results.

Drag and drop a file here or click browse to select a file.

2. Validate and Import into a Manual test run

Test Run Name

Test Run Started

Test Run Completed

Valid Test Case ID Title Test Point Parameters Configuration Tester

Det var min top ti, jeg håber den måske har kunnet skabe lidt inspiration hos dig til hvordan du kan bruge Azure DevOps til at få styr på din test i en hverdag i en agil kontekst. Men det vigtigste er nok tip nr 2; leg og lær 😊