

Slide 1



Slide 2



In Scrum, the Product Increment is the sum of all the Items completed in the current Sprint plus all the Items completed in all the prior Sprints. It represents a new version of the Product, that meets our Definition of Done, and that the Product Owner could decide to release into production.

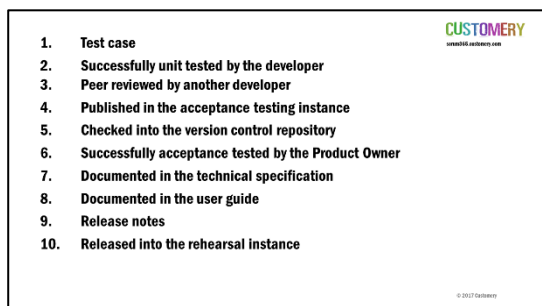
In my Dynamics 365 projects, in practical terms, that means we have a new solution package that the Product Owner could release into production if she or he wanted to. The changes inside the solution have been tested to ensure they meet the acceptance criteria for each Backlog Item and our project's Definition of Done that applies to all Items.

Slide 3



Done refers to a completed Backlog Item, and its Definition will vary from one Scrum project to the next. Some Microsoft partners have their own Definition of Done that they aspire to meet for all Backlog Items across all their customers' projects. Otherwise, the Scrum Team should agree a Definition of Done before Sprint 1. Things to consider including in your team's Definition of Done could include.

Slide 4



Backlog Items that are "Done" have

1. a test case
2. been successfully unit tested by the Developer
3. been peer reviewed by another Developer
4. been published in the acceptance testing instance
5. been checked into the version control repository

Product Increment

6. been successfully acceptance tested by the Product Owner or a proxy
7. been documented in the technical specification
8. been documented in the user guide
9. release notes
10. been released into the rehearsal instance

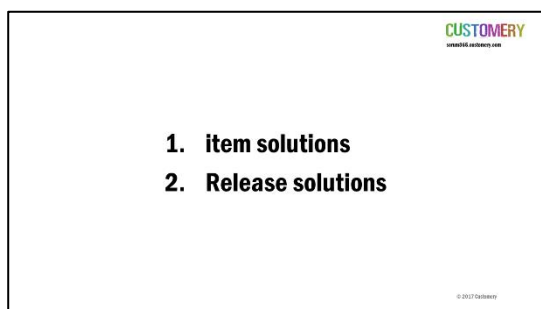
If Backlog Items must meet a lot of criteria before they are considered Done, you'll usually see the quality of your Dynamics 365 solutions improve, but unless you invest in devops automation then you might see the Scrum Team's velocity decrease. Typically, in my projects we start off with half of these criteria and add others as our velocity and level of automation improves.

Slide 5



Let's look at how Dynamics 365 solutions fit with Scrum's concept of Product Increments.

Slide 6



In my Dynamics 365 projects there are two types of Dynamics 365 solutions: item solutions and release solutions.

1. An item solution is a package containing the components that meet some requirements of a single Product Backlog Item. Usually the Item is a user story or a bug.
2. A release solution is a Product Increment; it contains all the solution components for all the Items in our release.

Slide 7

item solution naming convention

- **Item #**
- **Title**
- **Keywords**
- **Dev's name**

© 2017 Customery

I've found it helpful if the Dev Team has a solution naming convention for all their solutions. The item solution name includes

- The Item number from our Product Backlog
- Title of the Item using just a few words
- A keyword or tag or two, and
- The Developer's name or initials.

A structured naming convention makes it easy to search in the solutions view.

Slide 8

Solution: S0452 _ Sales funnel chart _ dashboards sales manager _ NeilB

Information

General

Display Name * S0452 _ Sales funnel chart _ dashboards sale

Publisher * Customery

Version * 1.0.0.0

Description

POST-RELEASE
Re-enable duplicate detection rules on Contact entity.

© 2017 Customery

Here's an example: S0452 _ Sales funnel chart _ dashboards sales manager _ NeilB

(In Dynamics 365 solutions must begin with a letter and an underscore is the only special character you can use.)

I also recommend using the Description field in the solution for describing any pre-release or post-release steps or a URL pointing to the Product Backlog Item in your requirements management system, usually Visual Studio Team Services or JIRA.

Slide 9

Unit testing

© 2017 Customery

The Dev Team can perform their unit testing in the dev instance, but it's better to if you can perform unit testing in a dedicated unit testing instance that you publish your item solutions into. Have a different Dev Team member, or even a customer team member, perform the unit testing.

Slide 10

Release solutions


© 2017 Customery

Once an item solution successfully passes its unit tests, I add its solution components to my current release solution. Then we import the release solution into the acceptance testing instance.

Slide 11

Release solution naming convention

- **Release #**
- **Sprint #**
- **Sprint goal**
- **Sprint end date**



© 2017 Customery


I've found it really helpful if the Dev Team has a solution naming convention for all their solutions. The release solution name includes

- A release ID
- Sprint #
- Sprint goal
- End date of the Sprint

The Description of the release solution includes the Item # and title of all the Items bundled inside.

Any Dynamics 365 Developers or MVP friends that want to build an XrmToolBox to help automate this procedure are welcome to get in touch.

Slide 12



Solution: R003 _ Sprint 3 _ Sales charts _ 20171206

Information


General

Display Name: R003 _ Sprint 3 _ Sales charts _ 20171206

Publisher: Customery

Version: 1.0.0.0

Description: S2462 Sales funnel chart
S2469 Sales velocity chart
S2465 Sales product mix chart
S2463 Regional sales manager dashboard
S2465 National sales director dashboard




© 2017 Customery

Here's an example: T003 _ Sprint 3 _ Sales charts _ 20171206

Release solutions are built in dev, exported from dev, and imported into your acceptance testing instance. Don't forget to also save it in your version control repository.

Slide 13

Production releases




© 2017 Customery

In this section we'll discuss my recommendations for releasing your Product Increment into production.

Slide 14

Minimum Viable Product



© 2017 Customery

My strong preference is to release into production at the end of every two- or three-week Sprint. In most projects, no one's going to use Dynamics 365 in production for the first few releases because it doesn't have enough useful features. Once it does have enough useful features to replace whatever the users are currently using, then our Product is known as the 'minimum viable product'.

Product Increment

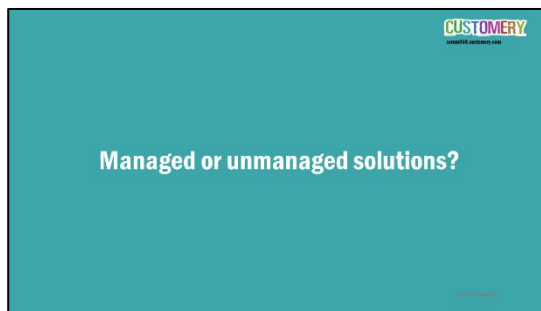
Slide 15



My preference is to keep releasing into production every Sprint after we've achieved MVP. Each release will deliver new or improved features for some of the users and usually a couple of bug fixes.

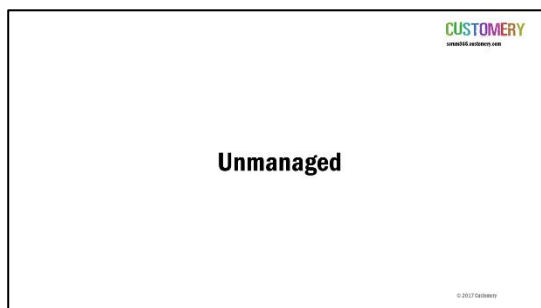
It's the Product Owner's job to determine whether to release each Increment into production or not. If every Increment impacts the same group of users every Sprint, then it might be too much change for the users to willingly absorb. But usually we can focus on different cohorts of users with every release so that the frequent releases don't become too overwhelming.

Slide 16



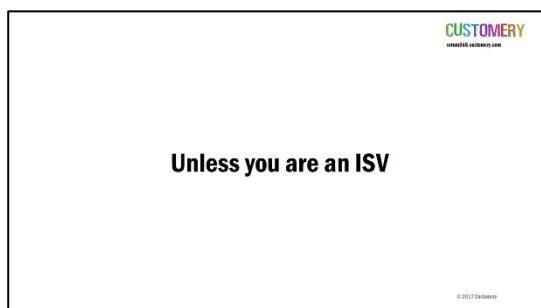
Should you use managed or unmanaged solutions?

Slide 17



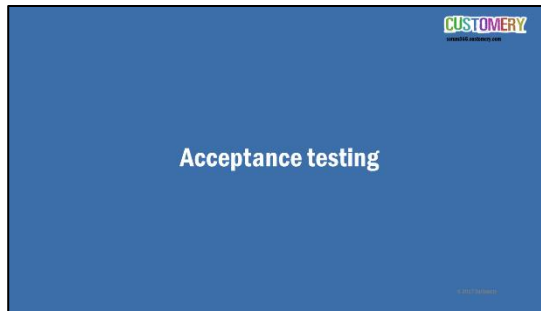
In my projects, I've always used unmanaged solutions to deploy features into production. While there have been several recent improvements in the solution engine, such as patches, that make managed solution files more useful than they used to be, I still haven't found managed solutions to offer enough advantages over unmanaged solutions. I recommend you stick with unmanaged solutions.

Slide 18



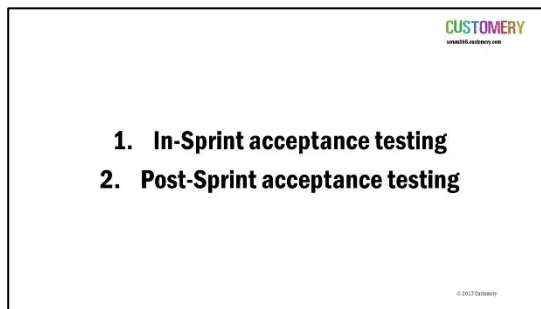
That's different, of course, if you are a software developer developing solutions that will be installed by lots of customers. Use managed solutions if you are an independent software vendor. But for everyday development by a Scrum Team working for one customer, whether you're a Microsoft partner or an in-house Scrum Team, use unmanaged solutions.

Slide 19



Let's discuss the two arrangements for acceptance testing on Scrum projects.

Slide 20



I've used two arrangements for acceptance testing in my Scrum projects:

1. In-Sprint acceptance testing – where the Items are tested, and usually accepted, within the same Sprint they are developed.
2. Post-Sprint acceptance testing – where the Items are tested in the Sprint after they are developed.

Slide 21



Acceptance testing within the Sprint requires that the Dev Team makes the feature ready for testing a couple of days before the end of the Sprint. The users need to be ready to test the features straight away and the Dev Team needs to have capacity to act on any feedback they receive to make the feature acceptable. I encourage teams to incorporate small changes within the Sprint, but the Dev Team decides what feedback they can incorporate straight away and what feedback needs to go back into the Product Backlog.

Slide 22



Acceptance testing in the Sprint after development gives the users more time to test the features. This sounds attractive to lots of customer teams, but it has a significant downside. While the acceptance testers are busy testing last Sprint's features, the Dev Team is busy in the current Sprint developing new features. So the Dev Team has no capacity to act on the tester's feedback. Every piece of feedback has to go into the Product Backlog. The result is that it can be several Sprints before a feature is developed, tested, refined, retested and eventually accepted. One of the goals of Scrum is to reduce the time

between good ideas and great software. We should always be looking for ways to adapt our processes to reduce this gap.

Slide 23



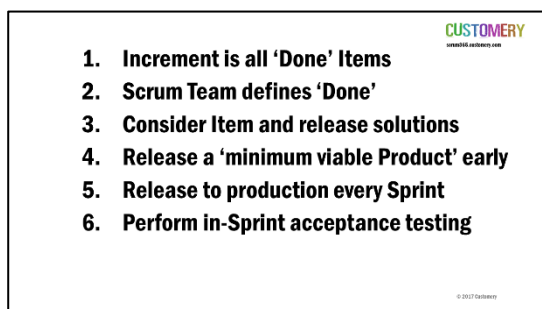
I always recommend acceptance testing within the development Sprint, and I never recommend acceptance testing after the Sprint. In-Sprint reduces the gap between idea and working software two or three Sprints quicker than testing after the Sprint. It requires focus and a sense of urgency, but it's worth it. Satisfaction guaranteed, or your money is wasted.

Slide 24



Let's recap what we've learned about the Product Increment.

Slide 25



1. Scrum defines the Increment as the sum of all Backlog Items completed in the current and prior Sprints that could be released into production.
2. The Scrum Team Definition of Done, which is the criteria that Backlog Items must meet before we consider them ready for release to production.
3. I use item solutions and release solutions, and I'd encourage you to explore this pattern too.
4. Release a 'minimum viable product' early in your project. Ideally, it'll be used by real users in production, so you can get quick wins and real feedback.
5. Refine your procedures so that you can release to production at the end of every Sprint.
6. I strongly encourage you to perform acceptance testing during the current Sprint, and not in a later Sprint.

Product Increment

Slide 26



Neil Benson
@customery
[Linkedin.com/in/neilbenson](https://www.linkedin.com/in/neilbenson)

Scrum for Dynamics 365
scrum365.customery.com

We're in it to ship it!

CUSTOMERY
scrum365.customery.com

© 2017 Customery