

Quality control on xRM solutions created with Microsoft Dynamics 365





Johan van Leenen RCP
is a manager within KPMG
Advisory.
vanleenen.johan@kpmg.nl

It is important for organizations to impose requirements and standards on the quality and reliability of xRM solutions

The ability to scale and adapt to productivity trends can become one of the success factors of a business. These trends include transitioning to Cloud, allowing mobile and remote access to users, and participating in the CRM/xRM technology that can build a personalized and extended relationships with individuals and companies. With the Microsoft Dynamics 365 platform, Microsoft offers a flexible and adjustable solution for these trends. For organizations using the Microsoft Dynamics 365 Platform and opting to make adjustments and additions to the Platform, it is important to take additional side effects into consideration. An important point of attention here is the quality of the solution to be created. The timely definition of quality standards and the monitoring of the quality during the execution of a project can become one of the important success factors in the development of a qualitative and future-proof solution.

INTRODUCTION

These days, cloud computing is increasingly maturing and growing at a rapid pace. The digital transformation that can be brought about by platforms and software solutions that are made available via the Cloud should be part of the CIO's agenda. There are many cloud solution providers. According to Gartner [GART19], Microsoft is one of the leaders in this field. Microsoft Dynamics CRM 2011 was their first step in offering CRM Cloud Services. Back in the day mostly small and mid-sized organizations saw the benefits of a Cloud-hosted CRM solution and the additional possibilities of tailoring relationship management to their needs. Since the introduction of Microsoft Dynamics 365, Microsoft's business applications have been increasingly embraced by organizations from the mid-market and larger business segment. As a result, projects have become larger and solutions more complex. Risks such as data leaks or GDPR violations – which these organizations are increasingly exposed to – and their consequences therefore generally become greater. For this reason, it is key for organizations to impose requirements and standards on the quality and reliability of, among other things, xRM solutions.

WHY MICROSOFT DYNAMICS 365?

With Microsoft Dynamics 365 for Customer Engagement (CRM, hereafter: Microsoft Dynamics 365), Microsoft provides an xRM platform. Companies can typically achieve a large percentage of their business requirements using the out of the box functionality of Microsoft Dynamics 365. The extensibility of the Microsoft Dynamics 365 Platform provides consultants and developers the possibility to configure and/or build custom line-of-business (LOB) applications that supply the remaining percentage of a company's requirements. To manage this xRM functionality, solutions were introduced in version 4.0 of Microsoft Dynamics. A solution is a package of all components of a project which is created and managed by either an ISV or an implementation partner. Through this process, additional functionality is created on top of the out of the box functionality of Microsoft Dynamics 365. Solutions also make it possible for ISVs to offer hosted (managed) solutions to end customers.

An independent software vendor (ISV) is an organization specializing in developing and selling custom software which focuses on a specific industry and solution. In this case, ISV solutions are designed as an addition to the out of the box functionality of Microsoft Dynamics.

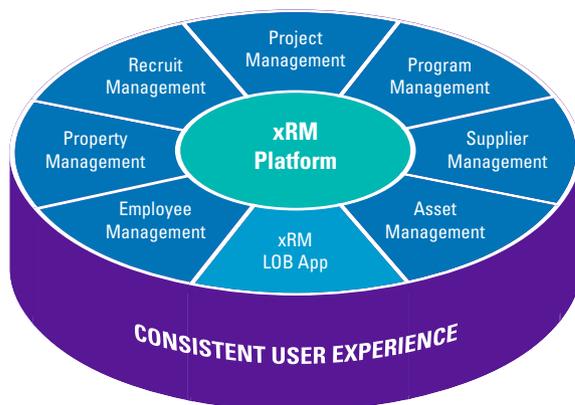
WHAT IS XRM?

Software developers and business analysts usually define xRM in one of two ways. In the first xRM definition, the 'x' in xRM refers to *extended* relationship management, which represents the extension of CRM platforms beyond customer relationship management. In the second xRM definition, the 'x' is a variable that can represent almost any relationship that a business needs to manage ([XRM19]). In both definitions, expanding the functionality of the CRM platform is the key. These xRM definitions mean that xRM systems can manage more than relationships with customers regarding sales, marketing, and service processes. It can manage anything a company might wish to manage in a relational database as depicted in Figure 1. For this reason, xRM depends on extensibility of the CRM platform on which it is built.

The 'x' from xRM: extending line-of-business (LOB) applications

In current projects there is limited time to write large amounts of custom code to deliver solutions. To meet requirements for business applications, you need a framework that provides the agility and flexibility to rapidly adapt to changes and get user acceptance and adoption. The way in which the Microsoft Dynamics 365 Platform facilitates in this regard differs from the way in which 100% custom solutions are setup. For quality controls this means that more than just code quality must be taken into consideration.

Figure 1. xRM provides the capability to build multiple LOB applications on a common platform with a familiar and consistent user interface ([IGN19]).



STANDARDS FRAMEWORK FOR QUALITY

Quality has many definitions, such as described in ISO/IEC 25010 [ISO14] (see the box on page 52). Key aspects of quality can include its maintainability (among other things due to the degree of duplication, unit size, complexity and modularity), code volume, test coverage, quality of documentation, and non-functional aspects such as security and performance.

To manage the expectations from both client as well as the suppliers, it is important to come to an agreement regarding the quality of the project to be delivered both prior to the realization of the project and for any further development phases. A Standards Framework for testing and maintaining the quality of the solution can be drawn

up to facilitate in this need. The following three steps provide insight into how a Standards Framework is created, how the Standards Framework for quality can be applied, and how it can be tested.

1. Establishing and managing the Standards Framework

There are no fixed guidelines regarding the quality standards that an xRM solution must meet. The reason for this is that every company or project is able to set their own bar, which, depending on the subject, can be set higher or lower. In addition, there is always a budget dependency as each organization or project uses a different cost-benefit ratio. For example, to get to basic principles, KPMG uses the industry “good practices”. Then, the final Standards Framework is determined in agreement with the customer.

The Standards Framework must also be aligned with the development teams who are responsible for delivering the xRM solution and managing the service landscape. The development teams should be given the opportunity to make recommendations about the substantive topics of the Standards Framework. After having clarified which recommendations will and will not be adopted in the quality Standard Framework, the framework can be finalized. After alignment with the teams, the results can be submitted to the client and the framework can be formally put into use.

Any future changes of insight in the area of xRM and code quality may lead to proposals for changes to the quality standard. Implementing these changes can follow the same process as setting the standard.

It is also recommended to include a demarcation to the Standards Framework. A demarcation makes the scope of the framework clear to both the customer and the development team. In addition, information regarding applicable nuances to specific components or preconditions that the xRM solution must meet can be provided.

The quality standards of a Microsoft Dynamics-based xRM solution can be roughly divided into two areas, which are Customization/Configuration and Custom Code. Both Customization/Configuration and Custom Code have their own area of attention. In the following paragraphs, the most common areas are described in outline.

Areas of attention for Customizations & Configuration

Upgradability check

A check on the presence of functionality that makes upgrading to newer Microsoft Dynamics 365 versions difficult to perform. This check includes, for example, deprecated and unsupported functionality.

You need a framework that provides the agility and flexibility to rapidly adapt to changes and get user acceptance and adoption

Solution structure

The use and design of solutions in the entire development pipeline can have an impact on maintainability, roll-out, and upgrade possibilities. In the case of solutions, it is therefore important to look at the number of circular dependencies and functionalities that come back in multiple solutions, the dependencies of third-party solutions (managed, without available source code), whose supportability and upgradeability are questionable, and the number of superfluous entities and web resources in the solution without an explanation.

Workflows and Plug-ins

During the development of an xRM solution, a proliferation of workflows and plug-ins can occur. For maintenance purposes, it is important that the number of unused triggers is reduced to a minimum. This also applies to synchronous workflows, the number of long-term asynchronous workflows and plug-ins that can lead to a time-out, and workflows with a wait or repeat condition.

Forms and Views

Microsoft Dynamics 365 comes with several forms and views that enable the end user to consult the information within an xRM solution. Additional forms and views can be added to the xRM solution by means of configuration. Just as with Workflows and Plug-ins, the prevention of unused forms and views is key to keeping an xRM solution maintainable and well-organized.

Web Service Calls

When creating a webservice call, it is important that the consulted fields are being used. Also, when using ‘Locks’ on queries there is a comment available and, if possible, paging on results and the presence of a ‘Top’ for the purpose of limiting the number of results is added to the queries.

It is important to come to an agreement regarding the quality of the project to be delivered

Security roles and authorization setup

The complexity of security roles does not affect the maintainability of an xRM solution immediately, but in combination with a complex setup of business units, access teams, owner teams and share functionality, the complexity factor can quickly grow. For this reason, it is recommended that all choices regarding the security of the application should be explainable.

Deployment automation

Manual operations significantly increase the risk of failures during a deployment. For this reason, it is highly recommended to invest in an automated deployment pipeline, for example, by creating an automatic deployment pipeline that can roll out builds across the entire development line. Depending on the budget and complexity of a project next to solution deployments, topics such as updating master data or migration steps due to changes in data model can be added to the deployment pipeline.

Documentation

It often happens that the documentation during projects does not have a high priority on the list of activities that must be carried out within a project. However, the way in which a project is documented is often a reflection of the quality of a project. Documentation that is in order simplifies the insight into a project and the transferability of a project.

Areas of attention for Custom Code Quality

Implementing Code Reviews

At the start of the realization phase, you should determine how often source code reviews are going to take place. An automated quality system in which the source code is continuously checked and reported can be used. SonarCube is an example of such tooling. While a project is based on predetermined standards, deviations from the

standard should only be permitted with the approval of the software delivery and quality managers.

Limiting complex source code

To promote the maintainability of the software, methods and classes should not be too complex. Alignment regarding the norm for the cyclomatic complexity of individual methods should be performed in advance.

Restrict duplication of source code

To promote the maintainability of the software, there must be as little duplication of source code as possible. The starting point for reporting on the standard settings for duplication will have to be determined in advance. This starting point for duplication in source code can then be monitored by a quality system.

Reduce the size of the system

Limiting the total size of the solution also contributes to improving maintainability. It is therefore important to properly monitor the number of lines of code that make up the xRM solution, to validate it for relevance and, if possible, to keep it to a minimum. The number of lines of code can also be monitored by a quality system.

Limit the size of methods

Another aspect to promote the maintainability of the software is to ensure that methods are not too large and do not contain too many parameters. A method is a set of code which is referred to by name and can be called (invoked) at any point in a program. Here too, it is important to set a standard so that the quality system can monitor and indicate the scope of the methods if the standard is exceeded.

In-depth - Definitions

Lock: Lock is a mechanism to ensure data consistency. SQL Server locks objects when the transaction starts. When the transaction is completed, SQL Server releases the locked object.

TOP: The SQL SELECT TOP statement is used to retrieve records from one or more tables in a database and limit the number of records returned based on a fixed value or percentage.

Cyclomatic complexity: Cyclomatic complexity is a software metric (measurement) which measures "the amount of decision logic (complexity) of a program or a source code function". Simply put: the more decisions that must be made in code, the more complex it is.

2. Applying the Standards Framework

During a project, the development team will apply the aligned quality standard of the Standards Framework to all adjustments that are made for the development of the xRM solution. Testing certain aspects of the Standards Framework related to customization and configuration is largely a manual process, where tooling such as Microsoft Solution Checker can be used when possible. For the assessment of the code regarding xRM functionality, Microsoft Premium Code Review [MICSr6] or specific quality systems like SonarQube can be used. In both cases, the standard as laid down in the Standards Framework can often be used as a basis.

It is possible that solving a situation in which a specific guideline from the Standards Framework is not met can lead to a qualitatively insufficient solution. In this case, for certain standards from the framework, the 'explain'-rule can be set. This rule means that for each individual deviation of a specific guideline an explanation ('rationale') must be documented explaining why the deviation is not resolved. The goal is to keep the number of 'rationales' as low as possible. Monitoring and initial substantive assessment of 'rationales' is the responsibility of the development team.

3. Testing the Standards Framework

It is recommended to carry out a periodical test on both quality assurance of the xRM configuration and the custom code/source code. The goal is twofold:

1. assess the extent to which the solutions produced by the project in Microsoft Dynamics 365 meet the Standards Framework set by the (internal) customer;
2. determine whether the quality assurance process is still in order.

This way a validation check is done on how findings are managed and whether the deviations from the defined rules ('rationales') are reasoned to a sufficient extent. Making the report, including a summary of the areas tested against (see Figure 2) available to the (internal) customer, ensures transparency in the quality of the development team.

Technical Debt

When applying or testing the Standards Framework, there is a chance that findings will emerge. These findings may be considered technical debt.

Technical debt could be thought of as a property of the software that threatens the long-term usability and maintainability of the software. Think of high complexity, low test coverage, missing test types, and missing documentation.

Aspects	Norm	Observation	
Upgradability	Maximum of 3 findings	2 findings	Green
Solution structure	Maximum of 10 findings	31 findings	Red
Workflows and plugins	Maximum average percentage of 5% 0 (zero, A-rating)	0,75%	Green
Forms and views	Maximum average percentage of 3%	1%	Green
Webservice calls to Dynamics 365	Maximum average percentage of 5%	13%	Red
Security Roles and Authorization	Maximum of 2 findings	No findings	Green
Automation of deployments	Average score of maximum 3	Score 3	Green
Representative test data and sandbox instances	Average score of maximum 2	Score 1	Green
Documentation	Minimal 80% of required documents in place	Score 77,5%	Red

Figure 2. Example of a Standards Framework quality review outcome.

The presence of technical debt has an adverse effect on the quality of the end products. That said, however, the emergence of technical debt during a project is often inevitable. It is also possible that some of the technical debt already existed at the start of the project and may not be resolved. In all cases, it is wise to know what technical debt exists. In order to ensure that the existing technical debt does not persist and its increase through new technical debt is avoided, it is important to plan the reduction of technical debt systematically.

To be able to deal with the debt systematically, it is first necessary to record the debt. Besides registration, it is recommended to add indications of aspects, such as the origins of the technical debt, the nature of the technical debt, the size of the technical debt, and the impact of the technical debt, in case it is not (timely) redeemed.

In addition to registering the causes and characteristics of the technical debt, it is also recommended to define how to deal with the different types of technical debt. By defining the different solutions regarding technical debt, any ambiguity is therefore avoided at the moment of observation. Finally, taking the correct measurements based on the predetermined procedure is key to higher quality standards. In projects executed with Agile Scrum, it is up to the product owner to prioritize technical debt on the project backlog.

About Microsoft Dynamics 365

Microsoft Dynamics 365 is a cloud-based ERP and CRM enterprise system developed by Microsoft. In 2003 the 1.0 version was released. This version mainly focused on relationship management and email marketing. The 3.0 version, which was released in 2005, was the first version with relationship management, sales, marketing, and service functionality built in. In addition to the functionality of version 3.0, solutions were introduced in version 4.0. Currently Dynamics Customer Engagement/CRM is part of the Dynamics 365 label which was introduced in 2016. In the core, Dynamics 365 still supports sales, marketing, and service processes, but it has been extended with process support for project service automation and field service automation. Additionally, ERP – Finance and Operations functionalities, which were previously available under the Microsoft Dynamics AX label, are combined in the core of Microsoft Dynamics 365. Figure 3 provides an overview of all components that Microsoft provides for the delivery of an intelligent business cloud. It explains the depth of the Microsoft Dynamics solution offering.

Microsoft Dynamics 365 deployment options

In general, there are two choices when it comes to deploying Microsoft Dynamics 365 – Online or On-Premise. The online version is hosted in Microsoft's Cloud and is a completely turn-key solution. The On-Premise version allows/ requires you to deploy and set up the servers and applications yourself. In the context of quality control, one of the most important differences is how upgrades and updates in both variants are handled. Where the online version uses an upgrade window determined by Microsoft, for the on-premise variant, this window can be determined by the support organization. Upgrading or updating a dynamics solution with solutions or code that does not (yet) meet the set conditions from the Standards Framework can have consequences for the operation of an xRM solution. An online deployment requires a method in which deviations from the quality standard must be detected and remedied more quickly than with an on-premise deployment.

About SonarQube

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities on 20+ programming languages. SonarQube offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security vulnerabilities. SonarQube can record metrics history and provides evolution graphs ([WIKI19]).

About Microsoft Solution Checker

Solution Checker promotes higher-quality model-driven solutions by helping developers follow best practices when they customize and extend the Microsoft Dynamics Platform. By running the checker, you can get answers to various quality solution-related topics. You can then access an actionable scorecard that lists top solution issues. For each identified issue, the scorecard points to specific occurrences within the code or customizations where improvements may be required.

About Microsoft Premium Code Review

Whether you are in the process of implementing Microsoft Dynamics 365 or currently operating a Microsoft Dynamics solution, it is likely that the out-of-the-box configuration must be customized to meet complex and evolving business requirements. With the introduction of customizations there comes an increased risk to meet the project milestones, operational service level agreements (SLAs), user experience expectations, and supportability standards. The Microsoft Dynamics Premium Code Review service seeks to provide deeper insights and tailored guidance. In contrast to an automated code review, which validates specific implementations on a pass/fail basis, Microsoft Dynamics Premium Code Review service considers more abstract factors, such as design requirements, contextual clues, and observations made throughout the code base to assess the design rationale and effectiveness of the chosen design.

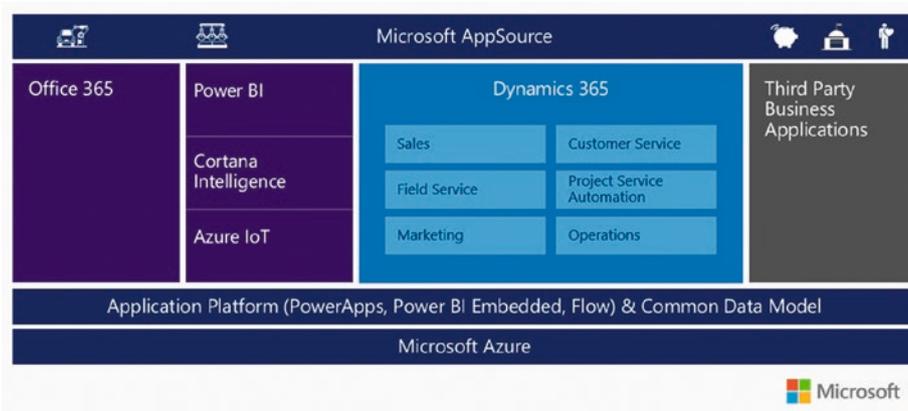


Figure 3. An overview of all the components Microsoft provides for the delivery of an intelligent business cloud ([Leit16]).

IN CONCLUSION

It is possible to create a high-quality xRM solution using Microsoft Dynamics 365. As an organization, before you start a project, think about which variant, Online or On-Premise, best suits your organization and solution.

Prior to the project, it is important that an organization thinks carefully about the quality standards that have to be met and the balance between the costs and benefits related to meeting a certain quality standard. It is advisable to draw up a Standards Framework for this, so that agreements on standards and the quality control during the development of an xRM solution and monitoring after completion and for possible further development processes are laid down. In addition, it is recommended to make agreements about how to deal with the technical debt that is built up when discovering findings.

It is possible to create a high-quality xRM solution using Microsoft Dynamics 365

References

- [GART19] Gartner. (2019). *Gartner Magic Quadrant for CRM*. Retrieved from microsoft.com: <https://info.microsoft.com/dynamics365-gartner-magic-quadrant-CRM.html?lcid=en-us>
- [IGN19] Ignify. (2019). *Microsoft Dynamics CRM Features*. Retrieved from ignify.com: http://www.ignify.com/microsoftcrm_XRM.asp
- [ISO14] ISO. (2014). *ISO/IEC-25000:2014 Systems and software engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE)*. Retrieved from iso25000.com: <http://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=3&start=6>
- [Leit16] Leite, R. (2016). *How partners can get ready for Microsoft Dynamics 365*. Retrieved from microsoft.com: <https://www.microsoft.com/en-us/us-partner-blog/2016/09/14/how-partners-can-get-ready-for-microsoft-dynamics-365/>
- [MICS16] Microsoft. (2016). *Microsoft Dynamics CRM Code Review: Premium*. Retrieved from: <https://msdnshared.blob.core.windows.net/media/2016/07/Dynamics-CRM-Code-Review-Premium-Datasheet.pdf>
- [WIKI19] Wikipedia. (2019). *SonarQube*. Retrieved from Wikipedia.org: <https://en.wikipedia.org/wiki/SonarQube>
- [XRM19] xRM. (2019). *Microsoft Dynamics CRM Blog, xRM defined by xRM*. Retrieved from xrm.com: https://xrm.com/xrm-2/xrm_defined/

About the author

Johan van Leenen RCP is a manager within KPMG Advisory. Johan is part of the Microsoft Team as part of KPMG's Enterprise Solutions division. He is a versatile Managing Consultant with a specialized focus on Microsoft Dynamics 365 Customer Engagement/CRM. Johan has over ten years of experience in implementing dynamics in roles as functional consultant, pre-sales consultant, project manager, product owner and trainer. Using his project experience with small and midsize clients in various sectors, he implemented CRM projects for more than twenty different customers. Johan is strongly focused and dedicated to system functionality optimization and usability/user-friendliness.