Artifacts for Agile Modeling: The UML and Beyond

This page provides a brief summary of potential models that you may choose to apply when developing business application software. This list is not complete, there are undoubtedly hundreds of types of artifacts available to you, but it does describe ones that are common use for modern software today. Furthermore, the focus is on the development of business application software not other types such as embedded software or system software. Yes, many of the artifacts could and should be applied in these other domains but the chart reflects a methodology geared for the development of modern-day business software. Finally, it is important to understand that you very likely won't need to apply every single technique on any given project, instead you will want to apply a subset that is appropriate to the task at hand.

The table below is large, sorry about that, and it may not contain everything that you need (such as a detailed description of the notation) which is why I include links to good references that describe the technique. For now you may want to refer to the Agile Models Distilled pages. Artifacts are listed in alphabetical order, links to detailed descriptions of each artifact are provided, and an explanation of the columns are at the bottom of the page. You may also want to refer to the article Be Realistic About the UML.

How does this chart support Agile Modeling (AM)? The principles *Multiple Models* indicates that you potentially need a wide range of models available to you, *Know Your Models* advises that you need to understand the strengths and weaknesses of each technique to apply them appropriately, and *Content is More Important Than Representation* implies that many artifacts have alternates that may be applicable for your situation. The practices *Create Several Models in Parallel* and *Iterate to Another Artifact* also require you to understand when and when not to apply a modeling technique, and *Discard Temporary Models* requires advice pertaining to when you should keep an artifact.

Artifact	Common Applications	Common Misapplications	Iterate To	Suggested Media	When to Keep It
Activity Diagram (UML)	Analysis or design of a business process or business rule	-	Class diagram Essential use case	Hand-drawn sketch Drawing tool	To provide a high-level overview of the logic for

	Design of the logic flow of a complex operation		Organization chart Source code System use case Usage scenario Use case diagram User story	CASE tool	business process
Business Rule Definition	Requirements identification	Documentation of technical requirements	Source code Class diagram CRC model Essential use case Flow chart System use case Usage scenario	Index card Word processor	When exact definition of business rul are required a stakehold readable format.

			Workflow diagram		
Change Case	Exploration of future potential requirements	Justification to overbuild software to meet "potential" requirements	Constraint CRC model Technical requirement Usage scenario Use case User story Workflow diagram	Index card Word processor	When you need to just design or architecture decisions to project stakeholders AND they require documentat
Class Diagram **(UML)**	Conceptual modeling	Physical database modeling	Activity diagram	Hand-drawn sketch CASE tool	You need to communicat the internal structure of

	Domain modeling Exploration of the structure of object-oriented software	Domain model documentation for users Only design diagram for OO software	Collaboration diagram Component diagram CRC model Data model Sequence diagram Source code State chart diagram Usage scenario User story		your softwa to others.
Class Responsibility Collaborator (CRC) Model	Domain modeling Conceptual modeling Exploration of the design of the structure of object-oriented software	-	Business rule Change case Constraint Class diagram Essential use case Organization chart System use case	Index card	Typically discarded a use.

			Usage scenario Use case diagram User story		
Communication Diagram (UML)	Exploration of the dynamic nature of complex object interactions	-	Class diagram Component diagram Deployment diagram Robustness diagram Source code System use case Usage scenario User interface flow diagram User interface prototype User story	Hand-drawn sketch CASE tool	Typically discarded a use May be kept show design of a complex portion of software
Component Diagram (UML)	Logical business		Class diagram	Hand-drawn	Often kept t depict high-

	architecture modeling Physical architectural modeling of a component-based software system		Deployment diagram Sequence diagram	sketch CASE tool	level architecture
Constraint Definition	Definition of a business or technical constraint	Definition of a Business rule Definition of a technical requirement NOTE: There's a fuzzy line between constraints and business rules as well as with technical requirements	Change case CRC model Deployment diagram Essential use case System use case Technical requirement Usage scenario	Index card Word processor	Kept as part official definition of requirement

			Workflow diagram		
Data Flow Diagram (DFD)	Analysis of existing business processes Design of new or updated business processes	Over specification of a system by "drilling down" into sub processes with more DFDs. Significant effort to level balance between a DFD and its sub-DFDs	Change case Constraint Data model Deployment diagram Organization chart Structure diagram System Use case Usage scenario Use case diagram User story Workflow diagram	Hand-drawn sketch Drawing tool CASE tool	To communicat overall desig of a process intensive system
				Hand-	

Data Model	Physical database design Conceptual or domain modeling for a data warehouse Explore relationships between a handful of entities	Conceptual modeling of OO software Domain modeling for OO software Exploration of structure of OO software A primary driver of the structure of a Class diagram	Class diagram Data flow diagram Deployment diagram Source code System use case Usage scenario User story Workflow diagram	drawn sketch CASE tool	To documer physical database design As a contrac model between the database owners and other syster accessing th database

Deployment Diagram (UML)	Identification of physical architecture for a system Identification of how software components are and/or will be deployed to physical architecture	?	Activity diagram Collaboration diagram Component model Constraint Data model External interface specification Sequence diagram Usage scenario User story Workflow diagram	Hand-drawn sketch CASE tool	To documer technical architecture your system
Essential Use Case	Identification of usage requirements for a system Identification of enterprise-level requirements for an organization	-	Change case Constraint Essential user interface flow prototype Robustness diagram System use case Technical requirement	Word processor CASE tool	Part of offici requirement documentat for a system

Essential User Interface Prototype	Exploration of the requirements for the user interface of a system	-	Business rule Constraint Essential use case User interface flow diagram User interface prototype	Paper (including Post-It notes) Hand-drawn sketch	Typically discarded
External Interface Specification (Contract Model)	Definition of interface (via an API, data feed, ...) to an external system	?	Data flow diagram Data model Deployment diagram Workflow diagram	Word processor CASE tool	As a contrac model between you system and external one
Features	Exploration of requirements	?	Acceptance test case Business rule definition	Index card Word processor	When you need a featu list describir your system

			Class diagram Class Responsibility Collaborator (CRC) model Collaboration diagram Constraint definition Essential user interface prototype Glossary Source code User interface prototype		
Flow Chart	Definition of complex logic	Over specification of logic when source code or specification language would do just as well	Class diagram Collaboration diagram Sequence diagram System use case Usage scenario User story	Hand-drawn sketch Drawing tool CASE tool	Typically discarded
Glossary	Definition of common terms	To much focus on getting it perfect.	Class diagram Data model	Word processor	Official definitions o terms.

			System use case Usage scneario		
Network Diagram	Analysis of existing technical infrastructure Design of proposed technical infrastructure	-	Component diagram System use case Workflow diagram	Hand-drawn sketch Drawing tool CASE tool	Official description technical infrastructur for your system or organizatior
Object Role Model (ORM) Diagram	Exploring domain concepts with stakeholders Conceptual modeling	-	Data model UML Class diagram	Hand-drawn sketch	Official conceptual model for yc system
Organization Chart	Depiction of existing or proposed	-	Activity diagram	Index cards & string	Official description the

	organization structure		Class Responsibility Collaborator Model Data flow diagram Use case diagram Workflow diagram	Hand-drawn sketch Drawing tool	organizatior structure of your enterprise o portion there
Package Diagram (UML)	High-level overview diagram that depicts the logical organization of requirements or a domain model High-level overview diagram depicting the physical organization of classes into packages To organize work, such as the assignment of requirements to specific subteams	-	Class diagram Use case diagram	CASE tool Index cards Hand-drawn sketch	When a CA tool diagram used to generate co into specific packages As an overview diagram depicting th organizatior requirement

Physical Prototype	Explore ergonomic issues of a system	-	Activity diagram Deployment diagram Network diagram System use case Usage scenario User story Workflow diagram	-	Typically discarded
Robustness Diagram	Analyze use cases to identify candidate classes and major user interface elements (screens, reports, ...)	To design user interface flow for a system To design static structure of OO software	Collaboration diagram Sequence diagram System use case Usage scenario User interface flow diagram User interface prototype User story	Hand-drawn sketch CASE tool	Typically discarded
Role Play	Exploration of the usage requirements for a system	-	Business rule Change case Constraint	-	N/A

	Verification that a system design will meet the needs of its users		Essential use case System use case Technical requirement Usage scenario User interface flow diagram User interface prototype User story		
Sequence Diagram (UML)	Modeling the logic of a usage scenario or a path though one or more use cases (or part(s) thereof)	Modeling of the logic for every single path through all the usage requirements for your system	Class diagram Robustness diagram System use case Usage scenario User story	Hand-drawn sketch CASE tool	Typically discarded

Specification Language (e.g. OCL)	Define precise logic of a process, operation, constraint, or business rule	Over specification on diagrams Detailed documentation for project stakeholders that likely don't understand the language	Business rule Class diagram Collaboration diagram Component diagram Dataflow diagram Workflow diagram	CASE tool Word processor	Part of your official definition of requirement
State Machine Diagram (UML)	Design the behavior of a complex class Analyze a complex business process	Design the behavior of several classes Model process flow Design the behavior of a simple class and/or one without interesting behavior based on state	Business rule Class diagram Source code System use case Usage scenario Table	Hand-drawn sketch CASE tool	May be kept as part of yo design documentat for complex class or process

Structure Diagram	Explore the "call" hierarchy within the design of procedural software	Design of object-oriented software	Dataflow diagram Source code	Hand-drawn sketch	High-level design of structured software
System Use Case	Analysis of usage requirements High-level design of implementation of usage requirements	Identification of usage requirements for a system The ONLY source of system specification for a system (e.g. you should avoid use-case driven [INSERT TERM HERE])	Collaboration diagram Essential use case Flow chart Robustness diagram Sequence diagram State chart diagram Usage scenario Use case diagram User interface prototype	Word processor CASE tool	Part of your design documentat for your system
				Word	

Table	Definition of complex business rules, constraints, or technical requirements	?	See business rules, constraints, & technical requirements	processor	Part of your official requirement definition
Technical Requirement	Requirements identification	Identification of business requirements Identification of "gold plate" requirements that the technical staff want to implement	Change case Constraint Deployment diagram Network diagram Workflow diagram	Index card Word processor	Part of your official requirement definition
Use Case Diagram (UML)	Overview diagram indicating major usage requirements Analysis of usage requirements of an existing system	Process diagramming Diagramming without supporting use cases	Activity diagram Data flow diagram Essential use case Organization chart System use case	Hand-drawn sketch Drawing tool CASE tool	Overview of your usage requirement

Usage Scenario	Exploration of the usage of a system	?	Activity diagram Business rule Constraint Deployment diagram Flow chart Network diagram System use case	Index card Word processor	Typically discarded

			Workflow diagram		
User Interface Flow Diagram	Exploration of user interface requirements High-level design of an application's user interface	?	Essential use case Robustness diagram System use case User interface prototype User story	Hand-drawn sketch Drawing tool CASE tool	Part of offici design documentat to provide overview of your user interface design
User Interface Prototype	Detailed design of a user interface	ONLY source of system specification Identification of user interface requirements ?	Business rule Constraint Essential use case Robustness diagram Source code System use case User interface flow diagram Workflow diagram	Hand-drawn sketch User interface prototyping tool	Typically discarded o evolved into working system
User Story	Exploration of usage requirements	?	CRC model	Index card Word processor	Typically discarded

	Reminder to have a conversation with a project stakeholder		Collaboration diagram Deployment diagram Robustness diagram Sequence diagram Source code Workflow diagram		
Workflow Diagram	Analysis of existing business process Design of new business process	ONLY source of system specification ?	Business rule Change case Constraint Deployment diagram Organization chart System use case Usage scenario User story	Hand-drawn sketch Drawing tool CASE tool	Part of your design documentat describing tl supported business processes

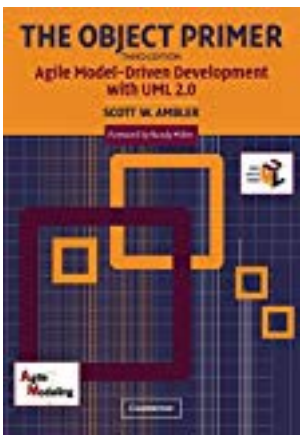
The Columns:

1. **Artifact**. The name of the modeling artifact.
2. **Common Application**. Common uses for the technique when developing business applications.
3. **Common Misapplication**. Common misuses of the technique that often lead to busy work or rework because the concept could be better captured using another technique. This column also includes advice for when not to apply the technique.

4. **Iterate to**. Suggested artifacts that are good choices to work on next from this one when following the practice *Iterate To Another Artifact*.
5. **Suggested Media**. An ordered list media which can be used to support the technique. Note: A drawing tool is something along the lines of Visio whereas a CASE tool is more along the lines of Rational Rose or TogetherSoft's Together.
6. **When to Keep It**. Advice for situations when it may make sense for this model to be a "keeper". However, never forget the principle of *Travel Light*.
7. **Likely Value as Keeper**. A rating of the likeliness that keeping the artifact will actually prove of value to your future efforts if you do decide to keep it.
8. **Alternate/Similar Artifact**. Modeling artifact(s) that are similar in nature to this one that could potentially be used as a replacement.
9. **Also known as**. Common aliases for this artifact.
10. **References**. Books or online resources describing the technique.

Recommended Reading

This book, Choose Your WoW! A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working, is an indispensable guide for agile coaches and practitioners to identify what techniques - including practices, strategies, and lifecycles - are effective in certain situations and not as effective in others. This advice is based on proven experience from hundreds of organizations facing similar situations to yours. Every team is unique and faces a unique situation, therefore they must choose and evolve a way of working (WoW) that is effective for them. Choose Your WoW! describes how to do this effectively, whether they are just starting with agile/lean or if they're already following Scrum, Kanban, SAFe, LeSS, Nexus, or other methods.



The Object Primer 3rd Edition: Agile Model Driven Development with UML 2 is an important reference book for agile modelers, describing how to develop 35 types of agile models including all 13 UML 2 diagrams. Furthermore, this book describes the fundamental programming and testing techniques for successful agile solution delivery. The book also shows how to move from your agile models to source code, how to succeed at implementation techniques such as refactoring and test-driven development(TDD). The Object Primer also includes a chapter overviewing the critical database development techniques (database refactoring, object/relational mapping, legacy analysis, and database access coding) from my award-winning Agile Database Techniquesbook.

DISCIPLINED AGILE

Agile Modeling

Agile Data

Enterprise Unified Process

@scottwambler