



The IDesign Method Management Overview

www.idesign.net

What is The IDesign Method

The IDesign Method (*The Method*) is a structured design methodology for producing the system architecture and the design of the project to build the system.

With system design, *The Method* lays out a way of breaking down a system into small modular software components used as building blocks. *The Method* offers guidelines for the structure and role of the components and how these components should interact, producing the architecture (high level software structure) of the system. *The Method* maximizes the ability to quickly respond to changes in the business (agility) by producing a small set of simple composable components that encapsulate changes. Through this, the architecture itself remains valid for a very long time without needing to change each time the requirements change. The result is minimizing the total cost of ownership, minimizing complexity, and maximizing quality.

With project design *The Method* provides management with several options for building the system, using the architecture as a key input for designing the project. Each option is some combination of schedule, cost, and risk. This allows decision makers to choose the best option, to drive educated decisions and to maximize the probability of the project successfully meeting its commitments.

As such, *The Method* is a two-part formula:

The Method = System Design + Project Design

The combination of system and project design offers strong competitive advantages for the business, compared with other design techniques that produce unmanageable, complex software, and that also offer no notion of time or cost. Unfortunately, the consequences of such out of control software are horrendous waste, longer projects, and frequent delays. Over time, such poorly-designed systems always end up with a prohibitive level of technical debt, leading to a re-write, that in turn often just repeats the mistakes of the past but on a larger scale.

Additional advantages of *The Method* include minimizing the required staffing level, lower turnover rate (people prefer working in a *Method* environment), low stress, high level of trust across the organization, improved customer satisfaction and retention, and reduced management risk.

The Method is used across hundreds of projects the world over, and has been in use for more than 20 years, with consistent repeatable results. In late 2019, Juval Löwy published a description of *The Method* in his book, [Righting Software](#) (Addison-Wesley Professional).

Audience

The purpose of this document is to inform business executives, business managers, project managers, product marketing managers, VPs of IT, CIOs and CTOs about the value produced by the IDesign Method, while avoiding diving into technical aspects. Software architects and software developers would benefit from reading this as well.

System Design Approach

Most systems today have a design and architecture that is characterized with severe technical debt, lack of ability to respond to changes, low quality, and inability to project time, cost, and risk. This is a direct result of the way most systems are designed, where the architecture reflects the required functionality of the system. Consequently, when the requirements change, the system architecture will also have to change, inflicting appalling cost and pain on all involved. When the design is based on functionality, the changes by definition are spread across multiple places in the system code.

To counter this behavior *The Method* prescribes designing the system based on volatility – identifying areas of change in the system, and encapsulating these in components. The required behaviors of the system are the integrations of these components. Now when the requirements change, the changes are contained and are not spread across the architecture and the existing software. Conceptually, the architecture is a series of vaults, where each of the vaults (as a component of the architecture) encapsulates some volatility. Now when a change happens, it is contained inside one of the vaults and there are no painful side effects and expensive software changes across the system.

Designing a moderate size system using *The Method* takes about a week for a trained architect.

Business Benefits

The obvious benefit of *The Method* is the ability to handle changes. During initial development and certainly during development of subsequent versions, requirements always change. Unlike designing against the required functionality, which maximizes the cost of the change (the change is spread all over the functionalities), volatility-based design minimizes the cost of handling the changes. You have contained the changes within components and are able to respond quickly to changes in the business landscape. That is the essence of agility that all businesses crave and all functionality-based designs always fail to deliver.

Structure and Communication

Most software systems share common areas of volatility. Furthermore, there are typical interactions, constraints, and run-time relationships between these common areas of volatility. By recognizing these common areas of volatility, the architect can produce the correct system architecture quickly, efficiently, and effectively. *The Method* provides classification for the areas of volatility and guidelines for the interactions between these areas. Having clear, consistent structure for components in the architecture and their relationship is not just a good starting point, but is essential for communication and ramp up time of developers, as well as speeding and smoothing the team's work along the project.

Project Design Approach

Most teams and projects have no clear idea upfront of the duration and cost of the project, or the risk involved. Therefore, decision makers are often flying blind. The only way to try and compensate is with wasteful and expensive overcapacity staffing and bloated estimations, which often makes matters worse.

With project design, *The Method* prescribes how to build the project as a network of development-related activities, and how to find the duration of the project. *The Method* instructs how to assign resources against the project network, and how to calculate the cost of the project. From the available time between activities in the network *The Method* also calculates the risk of the project using *The Method's* risk models.

Note that there is no single way of building any system. Projects designed to meet an aggressive schedule will cost more, be far riskier and more complex than projects designed to reduce cost and minimize risk. Project design narrows this spectrum of possibilities to several good project design options, such as the least expensive way to build the system, the fastest way to deliver the system, the safest way of meeting the commitments, and even the best combination of schedule, cost, and risk.

Since even a simple project can have several possible options that trade time, cost, and risk, *The Method* shows how to model these options and how to narrow it down to several good viable options for managers to choose from. Project design also sets up the project for simple project management and tracking.

Table 1 shows an example of the kind of options that project design can provide (from *Righting Software*, pp 360).

Design Option	Duration (months)	Total Cost (man-months)	Risk
Activity Driven	8	61	0.74
Architecture Driven	9	68	0.38
Understaffed	12	80	0.47

Table 1. Sample project design options.

Designing a decent size project using *The Method* takes about a week for a trained architect.

About Detailed Design

System architecture, as high-level design, is very different from detailed design. Detailed design focuses on the individual parts of a system, rather than the whole system. Detailed design is concerned with designing the structure of software code *within* each component and designing the interfaces that components use to interact (communicate) with each other. Detailed design is time consuming compared with architecture, and is not part of the design artifacts *The Method* produces.

It is important to note that just because *The Method* invests in a fast, up-front critical system and project design, does not imply it is like the Waterfall style of development with a “big upfront design”. Waterfall typically takes months to design everything to a minute level of detail before development can ever start. *The Method's* design is not “big”. Rather, while it is broad in scope, it is terse, yet good-enough to drive educated decisions up front, and a consistent, efficient design and development of the components later on. The details of the components such as the UX, UI, reports, data contracts, internal software design of each component, etc., are part of the development phase of the project, not its architecture. This sensible approach is very similar to the distinction between the architecture of a house from its specific details such as the structure of the walls, the furniture, and the paint.

Separating the work of system design from that of detailed design produces the following benefits:

- 1) To enable educated decisions, management only needs the output of project design (the options, such as Table 1), which in turn requires the architecture, but not the detailed design. Consequently, there is no need to invest in expensive detailed design in order to make these key educated decisions at the beginning of the project.
- 2) The team will perform the detailed design on the go during development, against the architecture, just when the details are needed. This allows keeping the team small, and the system within the confines of the architecture.
- 3) Given the right team composition (enough senior developers capable of detailed design), the project design can ignore detailed design since it is part of the development of every component.

Design Process vs. Development Process

By combining system and project design, *The Method* is a design process. As a design process *The Method* is independent of development processes such as Agile. This enables *The Method* to be used across any number of different specific development processes.

Business Value

Table 2 lists the range of business value *The Method* creates compared with previous generation system design techniques.

Business Value	Previous Techniques		<i>The Method</i>	
	System Design	Project Design	System Design	Project Design
System high level design, sooner or later	Maybe, Maybe not	Not supported	Sooner	Sooner
Unintended side effects	Yes	Not supported	No	No
Reduced time-to-market	No	Not supported	Yes	Yes
Reduced total-cost-of-ownership	No	Not supported	Yes	Yes
Ability to measure and manage risk	No	Not supported	Yes	Yes
An agile, low technical debt software code base that supports implementing requirements changes much faster	No	Not supported	Yes	Yes
Reduced waste	No	Not supported	Yes	Yes
Increased number and strength of competitive advantages to gain market share	No	Not supported	Yes	Yes
Potential asymmetric competitive advantages	No	Not supported	Yes	Yes
Increased probability of a successful project	No	Not supported	Yes	Yes
Increased quality	No	Not supported	Yes	Yes
Increased ROI	No	Not supported	Yes	Yes
Increased effective decision making	No	Not supported	Yes	Yes
Increased confidence in design methods used	No	Not supported	Yes	Yes
Increased confidence in development methods used	No	Not supported	Yes	Yes
Increased culture of effectiveness and success	No	Not supported	Yes	Yes
Increased team satisfaction and retention rate	No	Not supported	Yes	Yes
Increased innovation rate	Maybe at first	Not supported	Yes	Yes

Table 2. Comparing the business value of *The Method* with previous generation design approaches.

The Story Behind The Method

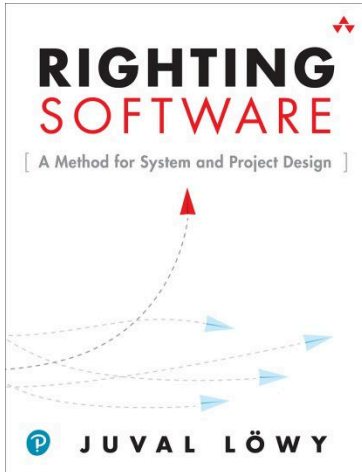
The Method was developed by Juval Löwy, founder of IDesign, and is a proven design process, rather than a new, idealistic approach. There is no manifesto or need to believe, and its track record speaks for itself. Löwy has trained thousands of practicing software architects the world over in the principles underlying *The Method* and how to use it.

Over the past 20 years Löwy and IDesign have successfully used *The Method* to design hundreds of systems for organizations of every type all over the world. Through this effort, IDesign has gained vast exposure to, and experience and understanding of, real world lasting system architectures. Indeed, there are few that have such breadth and depth of experience in real world system design and project design as IDesign.

The December 2019 publication of Juval Löwy's book [Righting Software](#), describing *The Method*, its principles and how to use it, opens *The Method* to the general public.

Resources

1. Righting Software, 1st Edition



By Juval Löwy, Addison-Wesley Professional, December 2019

2. The Architect's Master Class

The [Architect's Master Class](#) is the ultimate resource for the professional architect. The class shows how to take an active leadership role and is often referred to as a career-changing event. Alumni of the class are the architects of some of the most well-known companies and projects around the world. While the class shows how to design modern systems, it sets the focus on the 'why' and the rationale behind particular design decisions, often shedding light on poorly understood aspects. You will see relevant design guidelines, best practices, pitfalls, and the crucial process required of today's modern architects. Don't miss on this unique opportunity to learn and improve your design skills with IDesign, and share our passion for architecture and software engineering.

3. The Project Design Master Class

The [Project Design Master Class](#) presents our structured approach to project design and a comprehensive set of matching tools and techniques. You will master the steps, the interactions, the dynamics, the accurate modeling, the complexity reductions, the metrics, the rationale behind the intuition and experience. You will see how to perfect communication with top management, restore trust, and greatly increase your chance of success. The class provides guidance and knowledge that would otherwise take decades and many projects to acquire, and will propel your career like nothing else ever will.