February 3, 2010

Best Practices: Your Ten-Step Program To Improve Requirements And Deliver Better Software

by Mary Gerush
for Application Development & Program Management Professionals

FORRESTER® Making Leaders Successful Every Day

February 3, 2010

Best Practices: Your Ten-Step Program To Improve Requirements And Deliver Better Software

by Mary Gerush
with Mike Gilpin, Mike Gualtieri, Dave West, Tom Grant, Ph.D., and David D'Silva

EXECUTIVE SUMMARY

High-quality software requirements sit at the foundation of good software deliverables — those that meet business needs, satisfy customers, and fit into existing technology architectures. Yet many organizations haven't invested much (if at all) in their business analysts' development or requirements practices. If your organization is among these many, it's time to begin a requirements improvement journey. You can't afford *not* to do it, and a wealth of resources exists: research, educational materials, tools, and communities. Start by educating yourself and your organization on the importance of good software requirements and the cost of poor ones and teaching your business analysts what it takes to define and manage requirements well. Finally, adopt and adapt 10 requirements best practices that we've gleaned from research and discussion with enterprises, vendors, and industry thought leaders. Through improved requirements, you will make better use of scarce resources and satisfy — and perhaps even *delight* — your organization's employees and customers.

TABLE OF CONTENTS

NOTES & RESOURCES

Forrester interviewed vendor and user companies, including Blueprint Systems, Capgemini, Compliance Automation, IRS, and Seilevel.

Related Research Documents

"Just Do It: Modernize Your Requirements Practices"
April 15, 2009

"Are Your Project Teams Living In 'Template Hell?'"
March 20, 2009

"Harness The Power Of Your Business Analysts"
December 9, 2008

"Find And Grow Great Business Analysts"
October 22, 2008

MEDIOCRE REQUIREMENTS COST ORGANIZATIONS TIME, MONEY, AND MARKET SHARE

Today, enterprise success is irrevocably tied to the deployment and support of effective business technology solutions, which start with a well-defined set of business objectives, followed closely by a set of clear, well-understood requirements. Whether the deliverable is a business application, Web site, re-engineered process, or back-end service, good requirements are critical to delivering a successful solution. Yet many organizations don't do a good job of defining and managing requirements, and as a result, they commonly deliver solutions that fail to meet business needs, jeopardizing the enterprise's ability to succeed against its competition or in its mission.[1]

Delivering The Wrong Software Is Often Worse Than Delivering No Software At All . . .

Because organizations rely on sound technology to satisfy customers, support innovation, and increase market share, delivering a poor product can have drastic effects on an enterprise's success:

- **Enterprises squander development time and money when requirements are faulty.** A recent IAG Consulting study indicates that enterprises with very low requirements maturity waste 39% of their development budget and 46% of their available time due to faulty requirements. They deliver fewer projects on time and on budget than do organizations with higher levels of requirements maturity.[2]

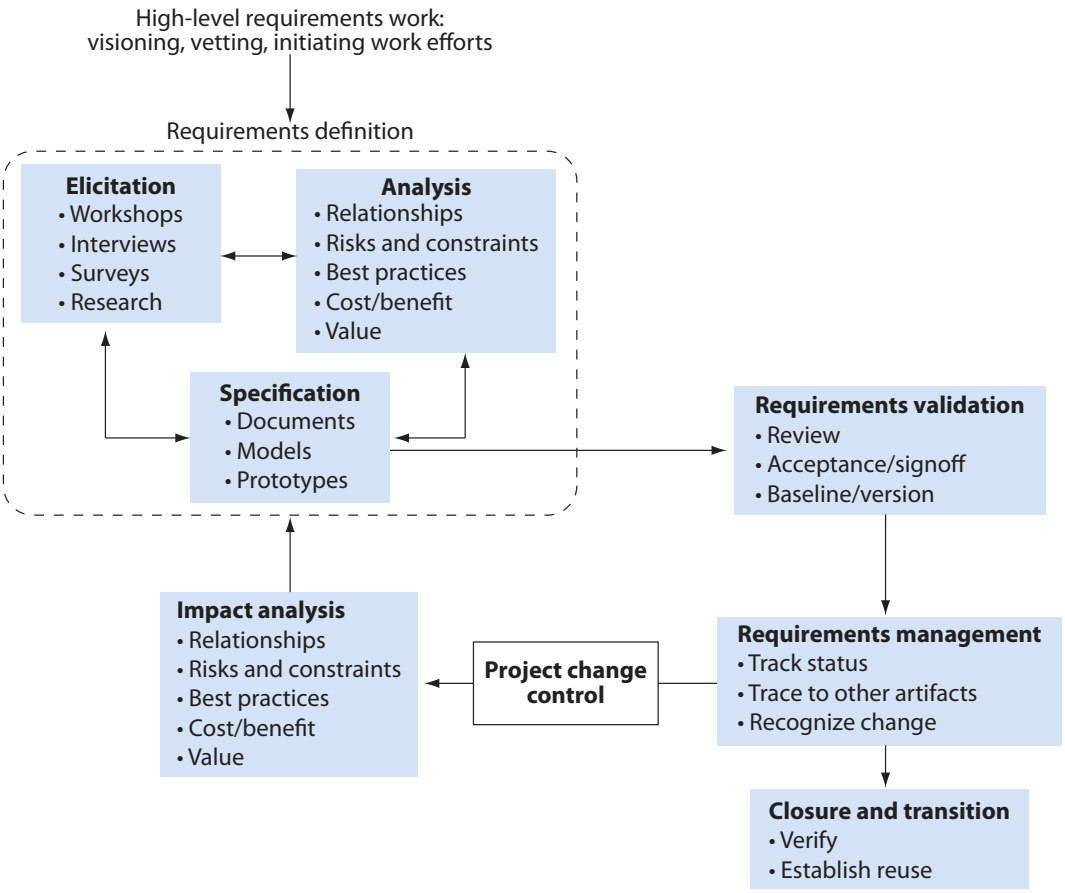
- **If your competitors get it right and you don't, your customers will move along.** If your Web site goes down, fails to function properly, or isn't as useful and usable as one of your competitors', you will lose customers.[3] Customers have choices; if you don't meet their needs, they'll check out your competition. You need to deliver more value to your customers than your competitors do, and high-quality technology solutions driven by effective requirements are critical to meeting this goal.

- **Poor software requirements affect organizations internally as well.** Internal business applications — such as finance systems and customer relationship management (CRM) applications — that don't adequately support business processes frustrate employees, commonly resulting in low adoption and the organization's failure to gain the system's intended benefits. Understanding your internal customers' software requirements is crucial to delivering the right solutions to meet their needs and keep the organization running effectively.

. . . But Getting Software Requirements "Right" Challenges Many Organizations

Whether called business analysts (BAs), requirements engineers, or one of a great number of other potential titles, the individuals responsible for defining and managing software requirements have tough jobs.[4] They must define high-quality requirements through the creative acts of elicitation, analysis, and specification and then manage them throughout the project life cycle (see Figure 1). Additionally, with business changing at light speed, requirements change frequently, so business analysts have to continually assess change's impact on requirements and help the requirements — and the software — evolve throughout the project. Requirements often aren't complete — or completely understood — until the solution has been delivered.

Figure 1 The Common Elements Of Software Requirements Definition And Management



55592 Source: Forrester Research, Inc.

Many project teams fail to define and manage requirements effectively, leading to stakeholders feeling that they didn't get what they wanted from the project's deliverables. And unfortunately, there is no easy answer; getting requirements "right" is challenging because:

- **Translating business needs into technology specifications isn't simple.** Business analysts work with many stakeholders — business partners, technology partners, customers, and end users — who often don't know how to express what they need from the system and frequently don't have time to participate adequately. To add to the complexity, software applications no longer sit in their own silos: They integrate with other internal applications and cross the firewall to talk with external systems. Many business analysts face difficulties finding the right people and extracting, analyzing, and communicating high-quality requirements effectively.

- **Business needs are moving targets.** In business, change happens. It's critical that the end result of a software project, particularly a lengthy one, meets the business needs that exist when it's delivered — not the year-old needs that existed at the start of the project. Therefore, a business analyst's work continues after initial project definition. She must maintain collaboration with the development and test teams, stay in sync with stakeholders, and be prepared to react to change, going back into requirements definition mode when new requirements arise or change affects existing ones. This view of requirements as an ever-evolving reality rather than a snapshot in time is particularly important when teams are using Agile and other iterative approaches to development and delivery.

- **Many business analysts have yet to master core skills . . .** For decades, business analysts have been an important part of the software delivery life cycle (SDLC), but until recently, there has been little understanding of the skills that they need and the practices that they use to deliver high-quality software requirements. Instead, they've been out there quietly doing their work with varying levels of success thanks to a lack of information and training; they've often focused more on documentation than on analysis and decision-making. Many business analysts don't know exactly *what* they should be doing, much less how to do it *well*, and many organizations don't know what they need from this important role. As a result, requirements maturity in many organizations is low. While a number of organizations, including a relatively new professional association called the International Institute of Business Analysis (IIBA), are working to educate business analysts and their companies on today's business analysts' high-level skills and best practices, many companies have a lot to learn and a long way to go to institutionalize business analysis and improve requirements maturity.[5]

- **. . . And today's environment demands that BAs understand more than the basics.** BAs need to enable their organizations to achieve faster software delivery while coping with increased technology complexity; therefore, they need to have more-advanced capabilities than in the past. Whereas 10 years ago they could get away with simply gathering requirements from stakeholders — like glorified order-takers — now they need to be consultants, negotiators, coaches, and decision-makers with a deep understanding of the business domain to help organizations invest scarce resources wisely. With global organizations increasingly prevalent, excellent communication and collaboration skills are more important than ever. And as business and technical complexity grows, BAs' analytical and critical-thinking skills must be top-notch.[6]

BEST DEMONSTRATED PRACTICES IN REQUIREMENTS DEFINITION AND MANAGEMENT

The bad news? *There is no checklist enabling business analysts to deliver high-quality software requirements.* It's a creative, analytical process that depends on a variety of stakeholders and high-level BA capabilities and is greatly influenced by organizational culture, industry, and project type.

The good news? Best practices in requirements definition and management have emerged and continue to mature. By educating your BAs and providing them with tools and techniques to use adaptively, you can help your organization improve requirements, deliver better software, and thrive.

Follow this 10-step program to improve the quality of your software through improved requirements:

1. Take a lean and mean(ingful) approach to requirements work.
2. Understand what good requirements look like.
3. Know all your stakeholders and their motivations.
4. Plan upfront using a tool kit approach.
5. Scrupulously tie requirements to business objectives.
6. Increase the use of pictorial requirements artifacts.
7. Do not neglect analysis!
8. Start with the big picture; iterate to a pragmatic level of detail.
9. Manage requirements throughout (and after) the project.
10. Adopt requirements tools that add the most value.

Best Practice No. 1: Take A Lean And Mean(ingful) Approach To Requirements Work

Lean is more than a buzzword; it's a "here and now" way of thinking that smart IT shops are adopting in all aspects of their work.[7] From a requirements perspective, Lean thinking encourages organizations to:

- **Eliminate waste in requirements processes wherever possible.** Gone are the days when stakeholders wanted or had time to read a 200-page Microsoft Word document to understand and validate software requirements. Today, they need just enough information in a friendly format — and no more. Similarly, project teams that spend most of their time in meetings — many of which they could avoid and replace with ad hoc conversations — are suffering the kind of waste that Lean thinking aims to eliminate. Look at your requirements definition and management processes with a critical eye, and get rid of anything that isn't critical to delivering the right software.

- **Design and create requirements deliverables based on value.** If your business analysts focus on creating perfectly formatted, template-driven, text-heavy documents to represent software requirements, it's time to change. In this "old world" environment, business analysts commonly create content that is unnecessary, redundant, and unread. How do you eliminate this waste? Only create artifacts that deliver concrete value. Ask yourself what your business stakeholders need to validate the requirements and what your developers and testers need to do their work. Don't create dozens of pages of text when a few screen mock-ups will do. And stop worrying about perfection. "We don't expect requirements to be perfect," says Martin Crisp, CTO, Blueprint Systems. "There is understanding and buy-in at all levels that 'this is enough; we get the gist and can go build something.'"

In all the interviews we conducted for this report, Lean thinking was a recurring theme that significantly influenced the best practices that emerged.[8] Educate yourself, your team, and your organization on Lean principles, and strive to apply Lean thinking across all aspects of software development and delivery.[9]

Pitfalls To Avoid When Applying Lean Thinking To Software Requirements

When putting on your Lean glasses, avoid:

- **Seeing "Lean" as a methodology.** Lean is an approach, a set of principles, and a way of thinking designed to eliminate waste and focus on value. Familiarize yourself with the ideals, but don't expect that through research you will find a "Lean" process designed to suit your organization. The benefit of Lean comes from applying Lean principals to your situation.

- **Thinking that eliminating waste means getting rid of process.** In all endeavors where multiple parties come together to create an outcome, some level of process is required. "Lean" doesn't mean "no process," "no documentation," or "no checklists"; it encourages organizations and teams to evaluate where prescribed processes, documentation, and checklists add value versus where they are overhead ripe for elimination.

Best Practice No. 2: Understand What Good Requirements Look Like

What are good requirements? The answer depends on the perspective of the stakeholder you ask. A number of audiences have a stake in the quality of software requirements, but they have varying needs and different definitions of "good." Effective business analysts know that to define good requirements, they must:

- **Elicit, analyze, and specify both *functional* and *nonfunctional* requirements.** The best requirements account for both functional requirements (those that indicate how the system should work) and for nonfunctional requirements (those that your architects, designers, security managers, auditors, and technical support staff care about *a lot,* such as security, performance, scalability, backups, and compliance) (see Figure 2). Where requirements are concerned, best-practice organizations look holistically at *all* stakeholders that the deliverables

will affect and expand their view of requirements to cover those stakeholders' needs. They also tailor requirements deliverables to each audience's needs.

- **Develop quality individual requirements as well as quality requirements packages.** Good individual requirements possess certain characteristics (see Figure 3). Smart organizations make sure that their business analysts understand how to create effective requirements in a variety of formats. These organizations also pay attention to the quality and completeness of the requirements package as a whole. Requirements packages should be thorough and well organized, conveying enough information to: 1) help stakeholders understand and sign off on requirements, and 2) help development and test teams understand business needs clearly.

- **Know that the proof is in the pudding: Successful outcomes are why we do the work.** Business partners don't care if a project's requirements documentation is perfect as long as its deliverables meet business needs and deliver value. Similarly, IT team members and other affected IT groups want quality outcomes that are maintainable and robust — not copious quantities of documentation. Successful project teams invest the most time in requirements activities that deliver the most value to the project team in the pursuit of delivering quality outcomes, and they measure the quality of the result as well as the process used to deliver it (see Figure 4).

- **Understand projects' context.** The work that requirements analysts do varies significantly with the projects they support. Requirements for a packaged software implementation are different from those for a custom-developed business application (see Figure 5). Requirements for externally facing apps are generally more rigorous and more focused on user experience than those for an application that only internal employees or partners will use. Business analysts must understand the context of the solution they're working on — its culture, dependencies, and audience — to know how to author "good" requirements.

Figure 2 "Good" Requirements Account For Both Functional And Nonfunctional Needs

Examples of functional requirement types	Examples of nonfunctional requirement types
• Adding information	• Security and privacy
• Modifying information	• Usability and design
• Deleting information	• Performance and response time
• Viewing and printing reports	• Capacity
• Integrating with other systems	• Stability, availability
• Processing data to deliver results	• Scalability
• Searching	• Backup and restore
• Purchasing goods	• Disaster recovery
• Logging on/off	• Compliance
• Creating user ID and password	• Quality
• Modifying user ID and password	• Release management
• Subscribing to email	• Documentation

55592 Source: Forrester Research, Inc.

Figure 3 BAs Should Strive For Quality Requirements Both Individually And Collectively

Component	Quality attributes	Measurement techniques
Individual requirement	• Unambiguous • Complete and correct • Current • Consistent • Observable and testable • Feasible • Mandatory and prioritized	• Peer reviews • Defect analysis (defects where root cause is a requirements defect)
Requirements package	• Complete • Clear, understandable • Tailored to various audience needs	• Survey of business stakeholders, designers and developers, and quality assurance team members • Defect analysis (defects where root cause is a requirements defect)

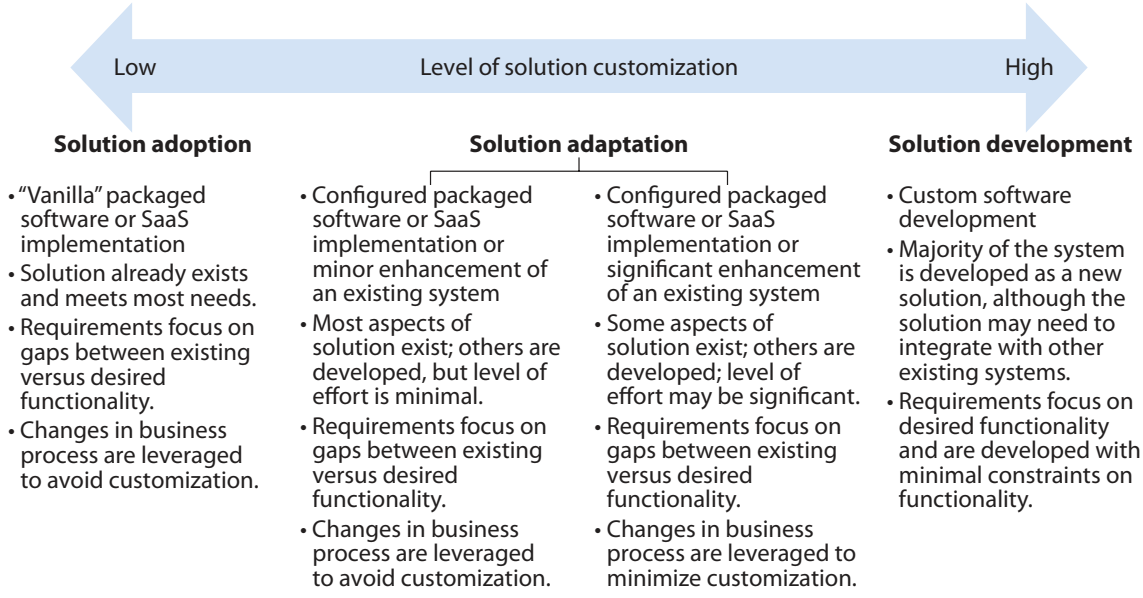
55592 Source: Forrester Research, Inc.

Figure 4 In The End, Successful Project Outcomes Are The Best Measure Of Quality Requirements

Component	Quality attributes	Measurement techniques
Product or other project outcomes	• Business goals met • High-quality product • Customers/stakeholders satisfied	• Customer/stakeholder satisfaction surveys • Quality measures • On-time, on-budget measures • Benefits realization analysis
Process	• Project team members satisfied • Customers/stakeholders satisfied	• Lessons-learned sessions • Project team member satisfaction surveys • Customer/stakeholder satisfaction surveys

55592 Source: Forrester Research, Inc.

Figure 5 Requirements' Goals And Approach Change Based On Project Type



55592 Source: Forrester Research, Inc.

Pitfalls To Avoid When Determining What "Good" Requirements Look Like

There is no single recipe for writing "good" requirements that you can apply to every type of technology deliverable. Best-practice organizations understand that and avoid:

- **Creating rigidity in requirements practices and deliverables.** Avoid rigid, prescriptive checklists and templates that dictate how your business analysts should approach their work. Instead, create flexible frameworks and tool kits for your business analysts to use on the types of projects they most commonly support. Provide a variety of templates — for example, gap analysis templates for packaged software implementations and wireframe templates for custom development projects. Teach your business analysts which templates they have at their disposal, and coach them on each template's best-fit situations.

- **Underexposing business analysts to a variety of conditions and project types.** Don't keep your business analysts siloed and limited to working on one project type. To design and manage software requirements effectively, business analysts need a good understanding of the project's environment. The more you expose them to a wide variety of business processes, technologies, and project types, the better they are able to know how to deliver "good" requirements that lead to successful outcomes. The best business analysts tend to have broad knowledge and experience across a variety of project types.

Best Practice No. 3: Know All Your Stakeholders And Their Motivations

Many business analysts think that the business is their only stakeholder, but new and changed software capabilities affect employees, customers, and business partners. Strong business analysts know that to be successful, they need to:

- **Unearth *all* the project's stakeholders.** While business sponsors and business staff on the project team are obvious stakeholders, there are many others. Enterprise employees use new and enhanced internal business applications, and the IT help desk supports them in that use. Business partners have to integrate reliably with systems. An organization's customers experience the impact of technology change, and call centers help those customers navigate the company's systems. Identifying *all* stakeholders is a critical first step to success.

- **Assess what's important to each type of stakeholder.** Each stakeholder group has its own objectives and motives. This is often the most complex part of successfully delivering project outcomes, and business analysts are at the forefront of understanding stakeholder needs and motivations to define and manage a quality set of requirements (see Figure 6). They also have to balance the needs of multiple stakeholder groups, pushing back on some requests in order to deliver the best overall outcome and the most business value. Understanding stakeholder motivations is crucial to successfully accomplishing this sometimes-difficult goal. Andrea Schneider, senior analyst, of the Internal Revenue Service says: "You have to have a good understanding of who the key players are. Who are your champions? Where are your anticipated roadblocks? You also *have* to have buy-in from IT."

Identifying all stakeholders and what they need from the solution gives a business analyst a much greater chance of identifying all functional and nonfunctional requirements.

Pitfalls To Avoid When Identifying Stakeholders And Analyzing Their Needs

It is important to understand who will be affected by technology changes, so savvy business analysts avoid:

- **Ignoring IT as a stakeholder.** One of business analysts' biggest shortfalls is thinking that they should focus only on the business. In reality, every change in technology affects a host of nonbusiness stakeholders — often significantly — including technology staff. A new installation of SharePoint requires internal technical support. An expanded data warehouse requires enhanced database and application support. IT is a stakeholder just as end users and customers are.

- **Overpromising to stakeholders.** Strong business analysts approach stakeholder analysis as an information-gathering activity: They relentlessly seek out stakeholders, assess their needs, and then look at the big picture to develop a realistic picture of the requirements that will meet those needs. They don't make early promises that the project team may not be able to keep; they set appropriate expectations as they interact with each stakeholder.

- **Compromising on ultimate business value delivery.** Developing and delivering new software requires tradeoffs, and delivering the highest business value matters most. While stakeholders all have their own best interests at heart, the best business analysts assess those interests as part of a larger picture, keeping a laser focus on the end goal — delivery of business value — and prioritizing and negotiating to create requirements that support that goal.

Figure 6 The Interests Of App Dev Project Stakeholders Vary

Stakeholder group	What they care most about
Company executives and governance body members	• Wise use of company resources • Business outcomes
Business sponsors	• Wise use of departmental resources • Business outcomes and customer satisfaction • Project success
Business project team members	• Project success
Business support and operations staff	• Ease of use and supportability
Technical project team members	• Project success
IT systems support and operations staff	• Technical fit and sustainability
Other organizational groups (legal, finance, operations)	• Compliance
Internal end users	• Ease of use • Functionality
Call centers and help desks	• Ease of use • Functionality • Understanding of systems changes
Third-party business partners	• System stability and availability • Ease of integration
Service providers and consultants	• Project success • Ongoing relationships
External customers	• Ease of use • Functionality

55592 Source: Forrester Research, Inc.

Best Practice No. 4: Plan Upfront Using A Tool Kit Approach

Dwight D. Eisenhower stated, "In preparing for battle, I have always found that plans are useless but planning is indispensable." The same is true when planning for requirements work as part of the software delivery process. But keep in mind that the word "planning" doesn't imply creating a rigid, rigorous process and a required set of artifacts. Successful business analysts enable success by:

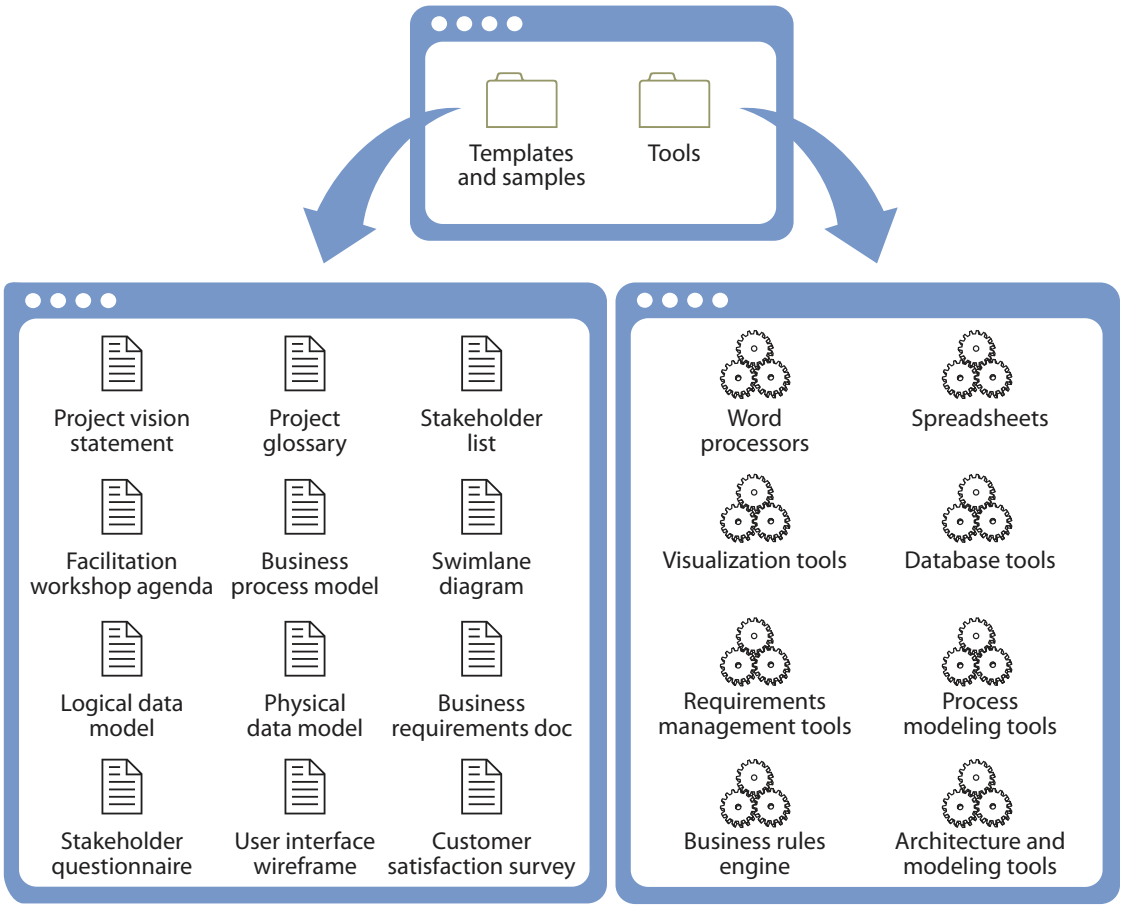
- **Planning the requirements approach early.** Business analysts don't just jump in without assessing how they will approach requirements work. They take time to understand scope and business objectives, identify and analyze stakeholders' preferences and needs, and define an approach to requirements definition and management. They plan how they will interact with stakeholders to define and validate requirements, choosing among techniques including interviews, requirements sessions, and surveys. They evaluate available templates, select ones they will use, and tailor them to their project. They determine if they will create prototypes and in what format. This is particularly important on larger projects where teams of business analysts will be defining requirements and where dependencies exist between the outcomes. In the end, requirements must fit together to deliver a complete view of the desired result.

- **Identifying a flexible set of relevant templates and tools.** Templates give business analysts a head start on requirements deliverables, and they facilitate consistency, which can help requirements consumers understand their content more quickly.[10] Likewise, tools help business analysts analyze, organize, communicate, and manage requirements. Strong app dev teams invest in a useful set of templates and tools, and they coach their business analysts in how to use and customize them to each situation, focusing on value (see Figure 7).

  There is no one "best" approach to creating requirements specifications, and there are a variety of models available, so make sure that your business analysts have an understanding of various models — such as process maps, use cases, user stories, and data models.

- **Planning to use the best techniques based on the audience and problem at hand.** In addition to templates and tools, business analysts rely on a variety of techniques, particularly when defining requirements. They facilitate workshops and design sessions, they design and conduct interviews and surveys, and they communicate in a variety of ways: through email, through collaboration platforms, and in discussions. Just as you don't need a sledgehammer to drive a nail, you don't need a formal meeting to gain the answer to one small question. Business analysts evaluate their stakeholders upfront and plan stakeholder engagement to optimize communication and eliminate waste. Ms. Schneider encourages "getting the most information in the least amount of time." Choosing the right techniques helps business analysts meet this goal.

Figure 7 Take A Flexible Tool Kit Approach To Requirements Templates And Tools



55592 Source: Forrester Research, Inc.

Pitfalls To Avoid When Planning Requirements Work

Planning is important, but at the same time, business analysts should have some freedom. Best-practice organizations try to avoid:

- **Prescribing one "de facto" approach to requirements.** By providing business analysts with a tool kit of templates, tools, and techniques, organizations empower them to use the skills for which they were hired. Smart organizations allow their teams and analysts to choose the right approach for their given situation.

- **Planning too late.** In any project, analysts will represent requirements in some format: Word documents, Excel spreadsheets, user stories, or entries in a requirements tool. For projects involving more than one business analyst, if the BAs fail to plan an approach to requirements

until late in the requirements process, they may have to rework requirements artifacts to achieve consistency. For very large projects where large teams of analysts are involved, the level of effort multiplies. By setting appropriate standards early, teams can avoid costly rework of requirements artifacts.

Best Practice No. 5: Scrupulously Tie Requirements To Business Objectives

Requirements exist long before projects begin; they start as ideas that turn into business strategies and objectives, which subsequently become the seeds from which projects and requirements grow. To increase the chances that end deliverables will meet business needs, best-practice software delivery teams connect requirements to strategies and objectives early and use them throughout the project to guide their work. To accomplish this important business-technology alignment, they:

- **Involve key business analysts early to focus them on business goals and objectives.** Many organizations wait to assign business analysts to projects until after initiation, missing the opportunity to provide them with a strong understanding of the project's importance and goals. Involving BAs after project initiation also means missing out on taking advantage of their analytical and critical-thinking skills, which can help in project scoping and planning. Assigning business analysts early — as ideas are evolving into projects — ensures that they gain critical understanding that serves as the foundation for good requirements and outcomes that meet business needs. Not *every* analyst needs to be involved early; for large projects, that's impractical. Therefore, assign one or more key analysts who can coach and guide others as they join the team.

  Some organizations leverage senior business analysts' talents *very* early — in strategic planning and portfolio management activities — recognizing that, through the work that they do on projects, business analysts very quickly grow to understand the business, the product, the customer, and the technology, making them powerful weapons in the war to deliver business technology value.

- **Use business objectives to validate requirements and eliminate waste.** "Gold plating" refers to a common problem that many project teams face. Requirements creep as stakeholders request new features — often "nice-to-haves" that are not critical to project success. Best-practice analysts link every requirement to a business objective to prevent scope creep, often including the business objective as an attribute of the requirement in their requirements tracking system. This focus on business alignment keeps the outcomes true to scope and tends to result in a simpler, more elegant product by preventing "bells and whistles" from obscuring mission-critical features.

  "You must have good business objectives to prioritize on your project. This helps you make a case for and deliver the majority of your business value. For example, if the business objective is to reduce costs by 20% over the next year, as a business analyst working with my

customer or business users to understand what features they need, I'm constantly holding that in my mind. When they give me something that I don't think aligns, I'll ask 'what are you trying to solve with that feature?' I'll try to dig out if there is really a connection there." (Joy Beatty, vice president of Blue Ocean Strategy, Seilevel)

- **Educate business analysts on the project's objectives as they join the team.** As new business analysts join, they need to understand the project's mission, goals, and objectives. Arming them with this information provides them with the foundation they need to contribute to project success.

Pitfalls To Avoid When Tying Requirements To Business Objectives

When defining and managing software requirements with the goal of business objective alignment, it's important that BAs avoid:

- **Assuming that business stakeholders fully understand and can communicate *what* they want.** Often, business stakeholders can't tell you what they need, or BAs hear something different from what stakeholders envision. Project managers and business analysts should prepare for the evolving clarity of objectives, continually questioning and refining their understanding over time. If business objectives remain unclear, this is a project risk that business analysts should identify, enabling the project manager to manage it through the project's risk management processes.[11]

- **Thinking that business objectives won't change during the project's course.** Change will happen, and as it does, business analysts need to continue to emphasize the connection between requirements and business objectives. Business objectives serve as the guidepost for validating and prioritizing requirements; should objectives change, teams should re-evaluate existing requirements against the new direction.

Best Practice No. 6: Increase The Use Of Pictorial Requirements Artifacts

Historically, business analysts have represented software requirements as words on paper, favoring Microsoft Word and Excel as their tools of choice. Increasingly, app dev teams and their stakeholders recognize the value of diagrams and models: Pictures convey information more quickly and generally take less time to create and update than their wordy equivalents. Many organizations have achieved an improvement in requirements accuracy and understanding by:

- **Increasing the use of process models.** Business stakeholders think in terms of business process. With the right facilitation, they can paint a picture of what they do every day and how they interact with people and systems. Business processes can be complex, and describing those processes textually can make certain process characteristics — such as branching, looping, and decision points — more difficult to digest (see Figure 8). Smart business analysts create process

diagrams — in the form of workflows, process maps, and swim lane diagrams — as a core part of their requirements packages.

- **Widely adopting the use of models such as context, use case, and data flow diagrams.** In addition to process models, business analysts use a variety of other diagram types. Context diagrams help them assess the impact of development work across systems and stakeholders. Use case diagrams provide information about the interaction of users and systems. Data flow diagrams show how information moves within and across organizations. Even organizational charts — which organizations don't commonly see as a requirements artifact — help business analysts validate that the team has identified all stakeholders. Ms. Beatty states, "We use org charts on all our projects to make sure that we don't miss stakeholders, actors, or big chunks of the organization."

- **Creating mockups, wireframes, and prototypes early in the project.** Not all projects entail visible user interfaces, but for the ones that do, business analysts should create screen mockups, wireframes, and even working simulations early in the project. "Sometimes describing what you want is more difficult than spotting it when you see it," says Tim Brown, Capgemini vice president, custom software. When project teams draw pictures or create simulations and walk users through scenarios, requirements elicitation happens naturally. Interestingly, although conventional wisdom says that one should avoid proposing solutions as part of the requirements process, this trend toward using mockups, wireframes, and prototypes brings some solution definition into the requirements stage by proposing experiences that help business users discover what works and what doesn't.

Pitfalls To Avoid When Increasing The Use Of Pictorial Requirements Artifacts

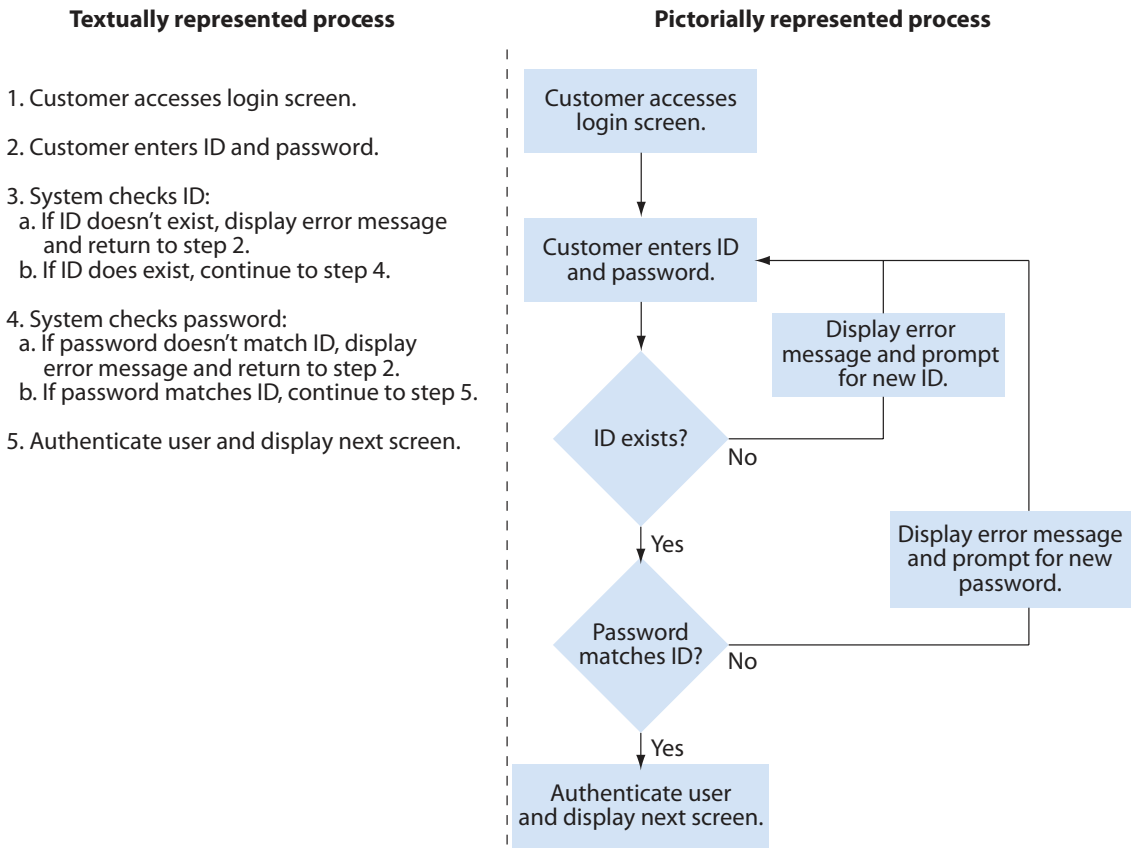
The trend toward increased use of diagramming, modeling, and prototyping in requirements is compelling, but it doesn't come without risk. Smart organizations avoid:

- **Getting bogged down trying to create perfect models and diagrams.** Analysts can rapidly become consumed by complex diagrams, wanting every detail to be perfect. This is especially true for prototypes and screen mockups, where the artifact is intended to closely approximate the final product. Train your business analysts to accept a "less perfect" style when creating pictorial artifacts to convey requirements or process.

- **Assuming that business analysts know how to create pictures.** Some people have a natural tendency to think in terms of pictures; others don't. When seeking to increase the use of diagrams, models, and screen mockups, assess your business analysts' abilities. You can bring many business analysts up to speed quickly through some basic user-experience training and a set of well-designed templates.

- **Thinking that every requirement can be represented pictorially.** A picture may be worth a thousand words, but words still matter — and as part of a model or wireframe, they are as important as the picture they augment. The combination of models and text is powerful and necessary to convey most aspects of software requirements.

Figure 8 A Simple Example Of Process Represented As Words Versus Pictures



55592 Source: Forrester Research, Inc.

Best Practice No. 7: Do Not Neglect Analysis!

Many requirements specialists have a title that contains the word "analyst," and yet many analysts fail to actually *analyze.* They ask business stakeholders what they want, document their responses, and ask for sign-off. This is order-taking, not analysis. Strong business analysts:

- **Know the business in their bones and how technology can help.** The best business analysts have a deep understanding of business — strategies, market trends, and customer needs in *their*

business — *and* of technology: what exists, what is feasible, and where opportunities for reuse exist. This knowledge provides a foundation for sound analysis that leads to valuable solutions.[12]

- **Analyze, consult, and negotiate with stakeholders.** The best business analysts use powerful analysis combined with strong communication and collaboration skills to navigate the murky waters of organizational and personal politics. The goal: satisfying stakeholders while balancing their needs and existing constraints and risks. Sometimes this requires business analysts to push back on business requests; other times, they must convince technology teams that a business concern is important enough to change the way they've done things in the past.

- **Make decisions.** Defining and managing successful requirements relies on business analysts' fundamental ability not just to listen but to listen, process, analyze, and consult with stakeholders. Business analysts must process all the information they gather to make decisions about which requirements tie back to business goals.

Pitfalls To Avoid When Analyzing Requirements

Analysis is difficult to define and specific to the situation at hand. BAs must avoid:

- **Analysis paralysis.** We've all experienced it. Often, a particular question does not have one clear answer — only many options, each with pros and cons — which can make it challenging to make the call and say "this is the way we will do this." Many business analysts revel in the analytical part of their jobs — it's what makes them good analysts. But at some point, they have to be able to move from analysis to making well-grounded recommendations and creating effective requirements deliverables.

  When analysis paralysis hits, find a way to "knock things loose." Give your business stakeholders something to review in the form of screen mockups or wireframes, or show them a few existing applications to demonstrate various approaches. Agile teams deliver working software at each iteration's end, allowing stakeholders to see and touch the project's deliverables early in the process. Traditional teams can follow this lead by creating rough prototypes as part of their process. It's often helpful to provide your stakeholders with a few options to spur creativity, confirm priorities, and move things along.

- **Neglecting business goals.** Do not let go of the end objective — delivery of business value — as the requirements evolve during the course of a project. Losing sight of the business objectives leads to requirements that may meet the needs of some stakeholders but that ultimately won't deliver the highest business value.

Best Practice No. 8: Start With The Big Picture; Iterate To A Pragmatic Level Of Detail

For decades, the waterfall methodology has prevailed: Project teams dedicate a significant amount of time early in the project to defining and baselining requirements. Increasingly, teams are learning that a more iterative, incremental approach to defining requirements delivers more value with less effort. These teams:

- **Start by defining requirements "a mile wide and 2 inches deep."** Agile teams do this naturally as part of their process. At the beginning of a project, they paint a picture with a very broad brush. Then, as they plan features for each iteration, they dive into the details and elaborate on requirements only enough to convey sufficient meaning for the team to develop and test the feature. Business analysts in traditional environments are recognizing the value in this approach and beginning to work from the top down, drilling into details where needed to confirm understanding. Ms. Beatty describes her BAs' process: "[They] start with high-level process flows and context diagrams, which lead them down the path to identify use cases and data flows. For more detail, they might use decision trees and data dictionaries, but they only create the lower-level models where they are relevant."

- **Elicit, analyze, and specify in short, iterative bursts.** Effective business analysts attack requirements in small chunks. They bring a small number of the right stakeholders together for requirements sessions, keeping the gatherings "short and sweet" and focused on a small piece of functionality. They go back to their desks and create a small set of models or wireframes or a prototype for stakeholders to review in another small gathering or in one-on-one meetings. Taking this approach enables stakeholders to see what they're asking for early and correct course if needed.

Pitfalls To Avoid When Starting With The Big Picture And Iterating To Pragmatic Detail

It's not easy to change the way project teams perform requirements work. When seeking to move to a more iterative requirements model, avoid:

- **Holding multiday requirements sessions with dozens of stakeholders present.** Large groups can't make decisions effectively, and stakeholders from a variety of groups will often use these large, long sessions to lobby for their own interests in front of the larger group. Avoid these large meetings; they can be a huge waste of time and aren't the most effective way to define or validate requirements. Instead, bring a larger group together to paint the big picture, then work with smaller groups and individuals to define the smaller chunks of functionality.

- **Underestimating the cultural change needed.** Stakeholders are used to multiday requirements sessions where everybody is in the same room making collective decisions. They are familiar with waiting until the end of the requirements phase to see the requirements and sign off on them in their entirety. Therefore, they may be reluctant to dedicate time for meetings and

reviews throughout the requirements process. Educating them on the value of this approach — and proving that value with better project outcomes — can go a long way toward facilitating this process change. If necessary, choose a project or two to pilot the approach and familiarize your stakeholders with the change.

Often, large requirements sessions and joint walkthroughs occur because there are trust issues in play. If multiple stakeholders have a stake in the same functionality, the project manager will want to make sure to involve all these stakeholders in every requirements session and walkthrough to obtain consensus. This is a tough challenge to overcome. In this environment, business analysts should work with the project manager to break up the work as much as possible and seek small stakeholder groups for requirements work.

Best Practice No. 9: Manage Requirements Throughout (And After) The Project

Once defined, requirements should continue as "living" artifacts: They drive development and testing and are realized in a work effort's deliverables. While requirements definition is an important and challenging phase of requirements work, managing validated requirements is also important. Best-practice organizations manage requirements by:

- **Baselining and versioning requirements through change control processes.** Today, business needs change frequently: New opportunities arise, leadership changes, and companies merge or go out of business. Organizations still have to bring projects in on time and on budget with high-quality deliverables. While ongoing requirements management is a moving target, with proper change control processes, requirements *can* evolve along with business needs and still maintain integrity. Smart organizations put a "stamp" on their baselined requirements but accept the inevitability of change. As change occurs, they promote strong change management processes and version requirements accordingly to ensure that metrics remain valid.

- **Assessing and managing the impact of change.** When business needs or technology environments change, business analysts in best-practice organizations spend time understanding how those changes affect the overall project and communicating that impact to project stakeholders. A changed business need can lead to new requirements, and a technology change can impose (or remove) constraints that alter the way users interact with a system. The best business analysts analyze change, assess its impact, and update requirements when necessary throughout the project life cycle. This discipline helps keep projects on track, improves the quality of end deliverables, and helps the team stay focused on delivering business value.

- **Tracing requirements to upstream and downstream development artifacts.** Traceability is not important in all environments. However, in some industries — particularly where regulatory compliance is critical — traceability plays a key role in requirements management because an auditor needs to use a certain set of test results to verify that an organization is meeting a compliance requirement. Many requirements tools facilitate linking and tracing requirements to other artifacts, but it's important to trace only where it adds value to the organization or project team.

- **Reusing requirements where reuse delivers value.** Business analysts should seek commonalities across functionality to create a set of requirements that organizations can use across projects and applications. In some industries, requirements reuse is fundamental. For example, many embedded systems are implemented with a set of alternative function variants to adapt the system to different applications or environments.

  In app dev shops, requirements reuse makes the most sense for nonfunctional requirements: security, performance, capacity, stability, and other common nonfunctional characteristics. If organizations can define standards for these requirements, they can more easily reuse them across projects, potentially simplifying the requirements process. Requirements for specific capabilities or services are also good candidates for reuse. For example, if app dev teams can establish commonalities across products and projects, they can reuse requirements for functions such as authentication, opting in and out of email distributions, and even generating reports.

Pitfalls To Avoid When Managing Requirements

Placing the right emphasis on requirements management is challenging. Best-practice organizations avoid:

- **Underemphasizing requirements management.** Underemphasizing requirements manage can lead to: 1) scope creep as new requirements emerge and the project team includes them in the system despite the fact that they haven't been analyzed and validated, and 2) dissatisfied stakeholders and end users when functionality fails to meet business objectives that changed after requirements sign-off. Inadequate requirements management can also jeopardize the outcomes of compliance reviews and audits if BAs do not create and manage traceability throughout the project. Releasing business analysts from projects after requirements definition puts project outcomes at risk, because change is inevitable, and project success depends on business analysts processing change's impact on requirements as the project evolves.

- **Placing too *much* importance on management activities that add no value.** Often, requirements management activities inject waste into software delivery processes but organizations feel compelled to do them because they are part of their standard process or methodology. If you can manage requirements change informally, do it — don't impose baselining and versioning unless your environment gains value in the form of requirements control and improved outcomes. And while requirements tools provide linking and tracing capabilities, use them wisely. Ron Tolido, Capgemini CTO, application lifecycle services, stated that traceability is important because "that is where it all starts. Requirements are supposed to be traceable." His team has also found, however, that sometimes traceability doesn't help ensure that it's meeting business challenges. "It then becomes more of an engineering challenge. You have to make sure that traceability doesn't become an autonomous 'mission in life.'"

