TABLE 15-2 **Traditional versus Agile Risk**

Risk Management with Traditional Approaches	Risk Dynamics with Agile Approaches
Large numbers of projects fail or are challenged.	Risk of catastrophic failure — spending large amounts of money with nothing to show — is almost eliminated.
The bigger, longer, and more complex the project, the more risky it is. Risk is highest at the end of a project.	Product value is gained immediately, rather than sinking costs into a project for months or even years with the growing chance of failure.
Conducting all the testing at the end of a project means that finding serious problems can put the entire project at risk.	Testing occurs while you develop. If a technical approach, a requirement, or even an entire product is not feasible, the development team discovers this in a short time, and you have more time to course correct. If correction is not possible, stakeholders spend less money on a failed project.
Projects are unable to accommodate new requirements mid-project without increased time and cost because extensive sunk cost exists in even the lowest-priority requirements.	Change for the benefit of the product is welcomed. Agile projects accommodate new high-priority requirements without increasing time or cost by removing a low-priority requirement of equal time and cost.
Traditional projects require time and cost estimates at the project start, when teams know the least about the project. Estimates are often inaccurate, creating a gap between expected and actual project schedules and budgets.	Project time and cost is estimated using the scrum team's actual performance, or velocity. You refine estimates throughout the project, because the longer you work on a project, the more you learn about the project, the requirements, and the scrum team.
When stakeholders don't have a unified goal, they can end up confusing the project team with conflicting information about what the product should achieve.	A single product owner is responsible for creating a vision for the product and represents the stakeholders to the project team.
Unresponsive or absent stakeholders can cause project delays and result in products that do not achieve the right goals.	The product owner is responsible for providing information about the product immediately. In addition, the scrum master helps remove impediments on a daily basis.

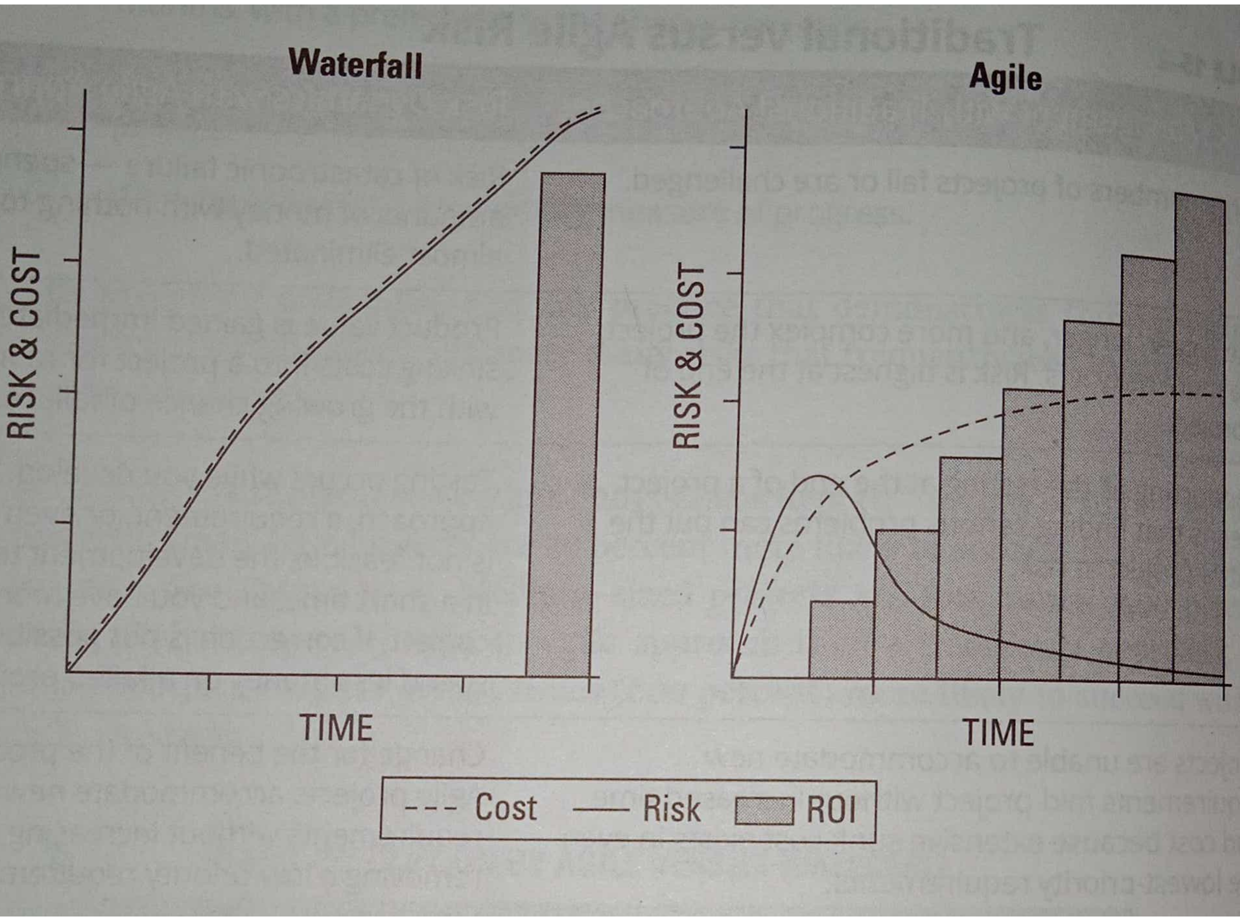

FIGURE 15-5: Agile projects' declining risk model.

Managing Agile Risk

Reducing risk *inherently*

Three especially important factors in risk reduction on agile projects are the definition of done, self-funding projects, and the idea of failing fast. You find out more about each of these factors in this section.

Risk and the definition of done

- **Developed:** The development team must fully create the working product requirement.
- **Tested:** The development team must have tested that the product works correctly and is defect-free.
- **Integrated:** The development team must have ensured that the requirement works with the whole product and any related systems.
- **Documented:** The development team must have created notes about how it created the product and the rationale behind key technical decisions made.

Figure 15-6 shows a sample definition of done, with details.

Self-funding projects

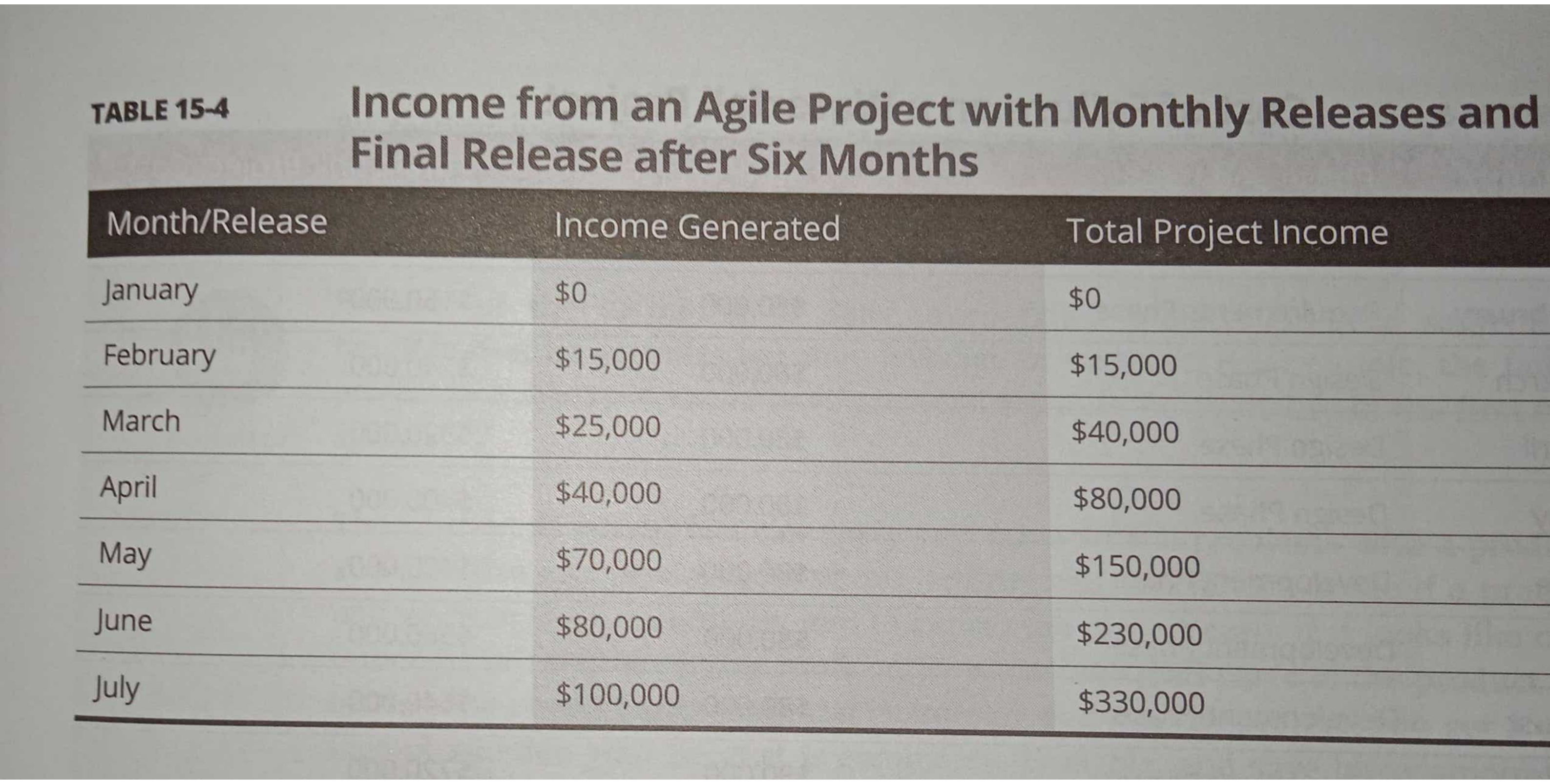

TABLE 15-4 **Income from an Agile Project with Monthly Releases and Final Release after Six Months**

Month/Release	Income Generated	Total Project Income
January	$0	$0
February	$15,000	$15,000
March	$25,000	$40,000
April	$40,000	$80,000
May	$70,000	$150,000
June	$80,000	$230,000
July	$100,000	$330,000

Failing fast

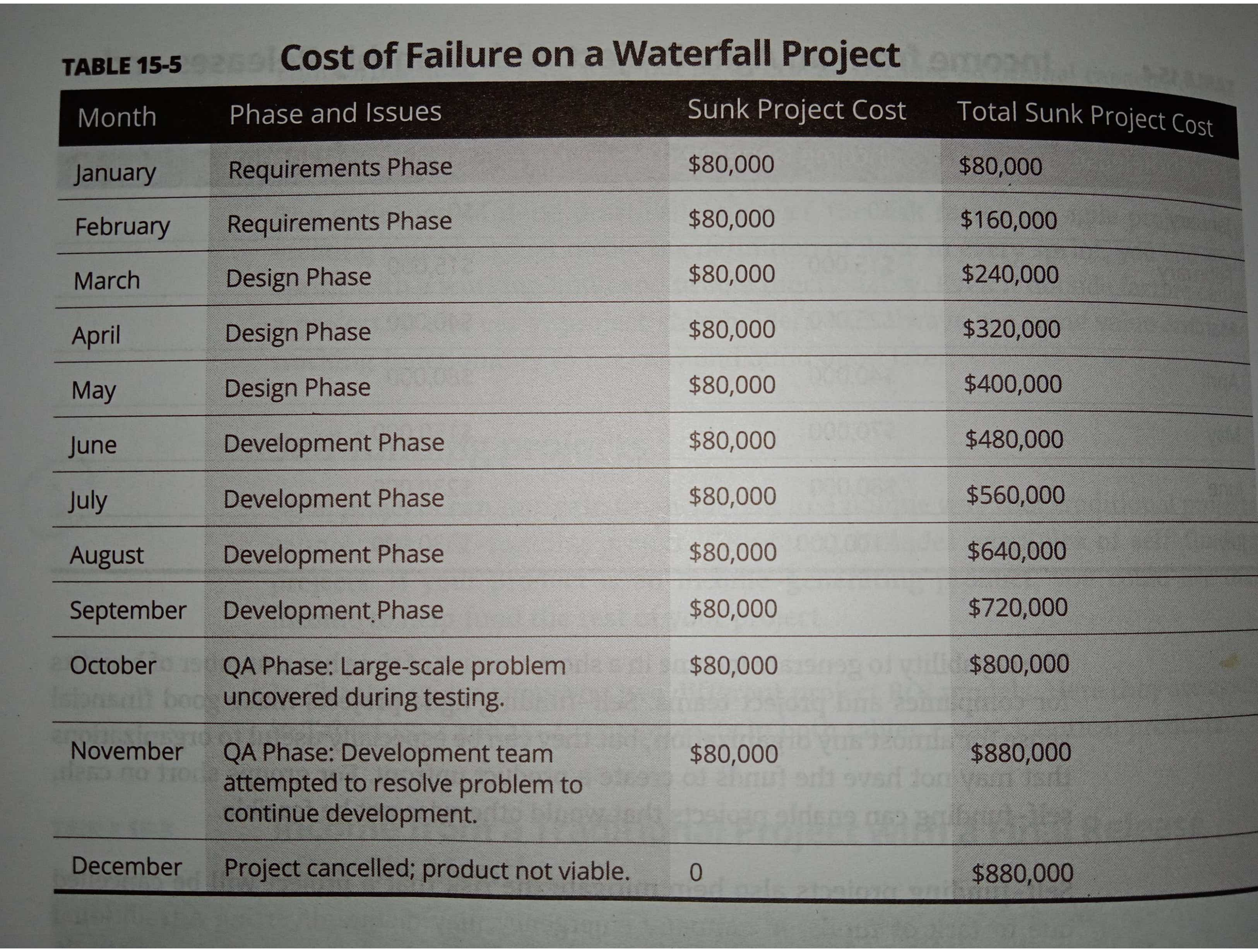

TABLE 15-5 Cost of Failure on a Waterfall Project

Month	Phase and Issues	Sunk Project Cost	Total Sunk Project Cost
January	Requirements Phase	$80,000	$80,000
February	Requirements Phase	$80,000	$160,000
March	Design Phase	$80,000	$240,000
April	Design Phase	$80,000	$320,000
May	Design Phase	$80,000	$400,000
June	Development Phase	$80,000	$480,000
July	Development Phase	$80,000	$560,000
August	Development Phase	$80,000	$640,000
September	Development Phase	$80,000	$720,000
October	QA Phase: Large-scale problem uncovered during testing.	$80,000	$800,000
November	QA Phase: Development team attempted to resolve problem to continue development.	$80,000	$880,000
December	Project cancelled; product not viable.	0	$880,000

TABLE 15-6 Cost of Failure on an Agile Project

Month	Sprint and Issues	Sunk Project Cost	Total Sunk Project Cost
January	Sprint 1: No issues. Sprint 2: No issues.	$80,000	$80,000
February	Sprint 3: Large-scale problem uncovered during testing resulted in failed sprint. Sprint 4: Development team attempted to resolve problem to continue development; sprint ultimately failed.	$80,000	$160,000
Final	Project cancelled; product not viable.	0	$160,000