Part 1
Introduction

Rationale research, which has been going on since the 1970s, initially focused on design rationale – the reasons behind decisions made when designing. This is an appropriate term in many domains where a physical artifact is first designed and then manufactured. While there is a phase in most software engineering (SE) lifecycles that produces a software design (design as a noun), the act of designing (design as a verb)—making the decisions that affect that design and how it is realized in the software system—takes place throughout the software development process. In order to make this distinction clear, in this book we refer to rationale as Software Engineering Rationale (SER), as defined in Dutoit et al. (2006b) and refer its use as a key aspect of the software process as Rationale-Based Software Engineering (RBSE).

The first step towards RBSE is an understanding of what rationale is and how it can help us meet the critical challenges that software engineering faces (Chapter 1). Software is not the same as hardware and these differences affect both what the rationale is (structure and content) and how rationale can be used (Chapter 2). These differences provide both opportunities, such as the ability to directly link rationale to the artifacts that it describes, and challenges, such as the need to support iteration.

SER can have many roles in supporting software engineering (Chapter 3). The decisions where rationale should be captured include not only those occurring during development but also those affecting the choice of software development process/methodology, management strategy, and how the software will be verified, validated, and even deployed.

The rationale research described here builds on work that started with Rittel's Issue-Based Information system (IBIS) (Kunz and Rittel 1970), initially applied to urban planning. Those proposing approaches to applying rationale to SE would be doing their research a disservice by not learning from the experience of applying rationale to other domains (Chapter 4). And finally, it is important to understand that decision-making, in particular human decision-making, lies at the heart of software engineering and how RBSE supports that process (Chapter 5).