For traceability links to be trusted by developers, they need to not only be created correctly, but also maintained so that they remain consistent and accurate. This is no mean feat in a typical software development project where it is the norm for requirements, design, code and test cases to constantly evolve to meet the changing needs of a project.

The first three chapters of this part of the book describe various approaches for dealing with traceability on evolving projects. The chapter by Mäder and Gotel focuses on maintaining “Ready-to-Use Traceability on Evolving Projects”. It promotes the use of a simple traceability information model, explores the requirements for automated maintenance and highlights a semi-automated approach. The chapter by Heider et al. details an event-based approach for acquiring and maintaining traceability links based on tracking changes to artifacts, “Evolution-Driven Trace Acquisition in Eclipse-Based Product Line Workspaces”, while the chapter by Seibel et al. discusses a rule-based approach to maintain model-to-model
traceability, “Traceability in Model-Driven Engineering: Efficient and Scalable Traceability Maintenance”. In a slightly different vein, and pertinent given the plethora of research on semi and automated techniques, the chapter by Dekhtyar and Hayes examines the role of the human to cast light on future directions for in-life cycle tracing research, “Studying the Role of Humans in the Traceability Loop”.