Software manufacturing is such a modern industry that there is a natural assumption amongst those that come into contact with it for the first time that newness equals advanced; as has been the reality with the hardware side. Far from experiencing a Moore’s Law for application software, disappointment has been a persistent experience amongst customers over decades concerning the interrelated factors of cost, flexibility, and less than optimal relations between IT and business teams.

The relatively low productivity of programming in general has had a major impact on business over the decades by driving extreme outsourcing and encouraging a wholesale shift to mass-produced, costly-to-modify products to keep the software affordable. This has lead in many cases to business processes and services having to be moulded to fit standardised software. In this way, the poor productivity of software creation and its subsequent modification has had both directly and indirectly damaging effects on business efficiency and costs.

This slowness of progress in the industry appears to arise in part from weak coalescing of thinking between the developer community, IT academia, and business/management consultancy. There has been surprisingly limited overlap or common working between them, and hence a difficulty in overcoming the huge drag created by ignorance and inertia amongst customers (risk of authorising change) and the power of vested interests, represented by the promulgation of different platforms, technology, and products with poor interoperability. There has been a gradual improvement over time, but the situation is still one of considerable confusion as to the best route forward, and all solutions proposed are partial (as the blogs and newsletters in the industry make clear).

It is surprising, but reassuring, to non-IT observers that all the components and methods necessary to build a methodology that is fully fit for purpose are in existence. This book explains how existing techniques and approaches can be combined to establish a comprehensive framework that permits developers to set down and change the specifications of business services easily and to automatically generate
comprehensive application software that fully reflects the nature of the provided services, integrating legacy software as necessary.

To achieve this objective, a holistic approach to the production process is required. Neither a full top-down business view or bottom-up technology view will do the job, nor can service provision be truly liberated through software applications when any limitation imposed by platform, technology, or language remains.

The key to this vision is the treatment of the organisation as a whole through a complete systemic view of all its constituent activities, actors, mechanisms, constraints, rules, resources, and external links, together with all their interactions, and the use of semantics to describe the complete business services model that everyone can understand. The vision is completed with transformation of this business model into platform-independent computer models and then into models specific to the desired e-services technology.

This revolutionary and universal systemic methodology for the rapid creation of enterprise applications, fully tailored to each business it covers, will empower companies to react quickly to changes in their environment or strategy and become totally independent in software implementation. Meanwhile, developers can become freed of the mundane task of programming and work closely with business analysts on modelling and designing higher level business services.

The economics of application software and the agility of business-decision implementation can be transformed.

_Graham Cox_

_Business Economist_