PART I  Introduction

1  Introduction
   1.1  The evolution of medical purpose software
   1.2  Product quality and software quality
   1.3  On the need for quality in medical purpose software
   1.4  Regulatory environments
   1.5  Verification and validation
   1.6  Structure of the book

PART II  Regulations

2  EU MDD 93/42/EEC
   2.1  Background
   2.2  Content of the Directive 93/42/EEC
   2.3  The approval process for software as a medical device
       2.3.1  Qualification
       2.3.2  Classification
       2.3.3  Selection of the Authorized Representative and notified body
       2.3.4  Implementation of a quality management system
       2.3.5  Documenting software as a medical device
       2.3.6  Auditing by the notified body
       2.3.7  Display of the CE marking

3  FDA title 21 of US CFR
   3.1  The role of the Food and Drug Administration
   3.2  Content of the Codes of Federal Regulation 21 CFR
   3.3  The approval process for Software as a Medical Device
       3.3.1  Qualification
       3.3.2  Classification
       3.3.3  Implementation of a Quality Management System
       3.3.4  Documenting the Software as a Medical Device
       3.3.5  FDA clearance and premarket approval

4 Regulations for other markets
   4.1 Regulatory environment and approval process in Australia
   4.2 Regulatory environment and approval process in Brazil
   4.3 Regulatory environment and approval process in Canada
   4.4 Regulatory environment and approval process in China
   4.5 Regulatory environment and approval process in Japan
   4.6 Regulatory environment and approval process in Russia

PART III Standards

5 ISO 13485: medical devices—quality management systems—requirements for regulatory purposes
   5.1 Introduction
   5.2 Contents
      5.2.1 The Quality Management System
      5.2.2 Management responsibility
      5.2.3 Resource management
      5.2.4 Product realization
      5.2.5 Measurement, analysis, and improvement
   5.3 ISO 13485:2016 versus other Quality Systems
   5.4 ISO 13485 certification
   5.5 Use of ISO 13485 in each jurisdiction

6 ISO 14971: medical devices—application of risk management to medical devices
   6.1 Introduction
   6.2 Contents
   6.3 Risk concepts applied to medical devices
   6.4 Examples of hazards, foreseeable sequences of events and hazardous situations
   6.5 Risk-management methods and tools
      6.5.1 Failure mode effects analysis
      6.5.2 Failure mode, effects, and criticality analysis
      6.5.3 Fault tree analysis
      6.5.4 Hazard analysis and critical control points
      6.5.5 Hazard operability (HAZOP) analysis
      6.5.6 Preliminary hazard analysis
      6.5.7 Markov analysis
   6.6 Use of ISO 14971:2007 in each jurisdiction

7 IEC 62304: medical device software—software life-cycle processes
   7.1 Introduction
   7.2 Content
      7.2.1 Software Development Process
      7.2.2 Maintenance process
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.3</td>
<td>Software risk management process</td>
<td>90</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Software configuration management process</td>
<td>92</td>
</tr>
<tr>
<td>7.2.5</td>
<td>Software problem resolution process</td>
<td>92</td>
</tr>
<tr>
<td>7.3</td>
<td>Use of IEC 62304 in each jurisdiction</td>
<td>92</td>
</tr>
<tr>
<td>8</td>
<td>IEEE 1012 and ISO/IEC 29119: standards for software verification</td>
<td>95</td>
</tr>
<tr>
<td>8.1</td>
<td>IEEE Std 1012 for system and software verification and validation</td>
<td>95</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Integrity levels</td>
<td>97</td>
</tr>
<tr>
<td>8.1.2</td>
<td>Common V&amp;V activities</td>
<td>97</td>
</tr>
<tr>
<td>8.1.3</td>
<td>Software V&amp;V activities</td>
<td>97</td>
</tr>
<tr>
<td>8.2</td>
<td>ISO/IEC 29119 software testing</td>
<td>99</td>
</tr>
<tr>
<td>8.2.1</td>
<td>ISO/IEC 29119-1: concepts &amp; definitions</td>
<td>100</td>
</tr>
<tr>
<td>8.2.2</td>
<td>ISO/IEC 29119-2: test processes</td>
<td>100</td>
</tr>
<tr>
<td>8.2.3</td>
<td>ISO/IEC 29119-3: test documentation</td>
<td>104</td>
</tr>
<tr>
<td>8.2.4</td>
<td>ISO/IEC 29119-4: test techniques</td>
<td>104</td>
</tr>
<tr>
<td>8.2.5</td>
<td>ISO/IEC 29119-5: keyword-driven testing</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td><strong>PART IV Verification and validation techniques</strong></td>
<td>107</td>
</tr>
<tr>
<td>9</td>
<td>Static testing</td>
<td>109</td>
</tr>
<tr>
<td>9.1</td>
<td>Introduction and background</td>
<td>109</td>
</tr>
<tr>
<td>9.2</td>
<td>Static testing</td>
<td>110</td>
</tr>
<tr>
<td>9.3</td>
<td>Static analysis</td>
<td>111</td>
</tr>
<tr>
<td>9.3.1</td>
<td>Control flow analysis</td>
<td>111</td>
</tr>
<tr>
<td>9.3.2</td>
<td>Data dependence analysis</td>
<td>114</td>
</tr>
<tr>
<td>9.3.3</td>
<td>Control dependence analysis</td>
<td>120</td>
</tr>
<tr>
<td>10</td>
<td>Dynamic testing</td>
<td>121</td>
</tr>
<tr>
<td>10.1</td>
<td>Introduction</td>
<td>121</td>
</tr>
<tr>
<td>10.2</td>
<td>Specification-based testing technique</td>
<td>122</td>
</tr>
<tr>
<td>10.2.1</td>
<td>Equivalence partitioning</td>
<td>122</td>
</tr>
<tr>
<td>10.2.2</td>
<td>Boundary value analysis</td>
<td>123</td>
</tr>
<tr>
<td>10.2.3</td>
<td>State transition testing</td>
<td>124</td>
</tr>
<tr>
<td>10.2.4</td>
<td>Cause–effect graphing and decision table testing</td>
<td>125</td>
</tr>
<tr>
<td>10.2.5</td>
<td>Syntax testing</td>
<td>127</td>
</tr>
<tr>
<td>10.2.6</td>
<td>Combinatorial test techniques</td>
<td>128</td>
</tr>
<tr>
<td>10.2.7</td>
<td>Scenario testing and use case testing</td>
<td>131</td>
</tr>
<tr>
<td>10.2.8</td>
<td>Random testing</td>
<td>132</td>
</tr>
<tr>
<td>10.3</td>
<td>Structure-based testing technique</td>
<td>132</td>
</tr>
<tr>
<td>10.3.1</td>
<td>Statement testing</td>
<td>132</td>
</tr>
<tr>
<td>10.3.2</td>
<td>Branch/decision testing</td>
<td>133</td>
</tr>
<tr>
<td>10.3.3</td>
<td>Condition testing</td>
<td>135</td>
</tr>
<tr>
<td>10.3.4</td>
<td>Data flow testing</td>
<td>135</td>
</tr>
<tr>
<td>10.4</td>
<td>Error-guessing testing technique</td>
<td>136</td>
</tr>
<tr>
<td>10.4.1</td>
<td>Error-guessing</td>
<td>136</td>
</tr>
</tbody>
</table>
### 11 Formal verification

- **11.1 Introduction and background**
- **11.2 Formal specification**
  - 11.2.1 Ambient calculus and ambient logic
  - 11.2.2 Linear temporal logic
- **11.3 Model checking**
- **11.4 Static and dynamic (formal) verification**
- **11.5 Summary**

### 12 Prescriptive software development life cycles

- **12.1 Software as a product**
- **12.2 Software development strategies**
- **12.3 Waterfall models**
  - 12.3.1 The waterfall
  - 12.3.2 The V-model
- **12.4 Evolutionary models**
  - 12.4.1 Prototype models
  - 12.4.2 The incremental model
  - 12.4.3 The spiral model
- **12.5 Choosing the best software development model**

### 13 Agile software development life cycles

- **13.1 The Agile Manifesto**
- **13.2 Scrum**
  - 13.2.1 Roles
  - 13.2.2 Events
- **13.3 Agile testing practices**
  - 13.3.1 Test-Driven Development
  - 13.3.2 Acceptance Test-Driven Development
  - 13.3.3 Behavior-Driven Development
- **13.4 Agile in a regulated environment**

### 14 Project management

- **14.1 Introduction**
- **14.2 Initiating**
- **14.3 Planning**
  - 14.3.1 Setting the goals
  - 14.3.2 Assigning the responsibilities
  - 14.3.3 Defining the scope
  - 14.3.4 Planning time and costs
- **14.4 Executing**
14.5 Monitoring and controlling 186
14.6 Closing 187

15 Risk management 189
15.1 Risk assessment overview 189
15.2 Risk assessment workflow 192
15.3 Static versus dynamic safety risk scenarios 196
15.4 Probabilistic risk model 199
15.5 Application to the case study 200
  15.5.1 Safety critical factor identification 200
  15.5.2 Risk analysis 201
  15.5.3 Risk scenario development 202
  15.5.4 Probabilistic risk model 204
  15.5.5 PRM analysis and risk evaluation 206

16 Requirements management 209
16.1 Background 209
16.2 Types of requirements 210
16.3 Requirements development 213
  16.3.1 Requirements elicitation 214
  16.3.2 Requirements specification 214
  16.3.3 Requirements verification and validation 217
16.4 Requirements traceability 217

17 Design controls and development management 219
17.1 Background 219
17.2 Design controls 220
17.3 Design control and development templates 221
  17.3.1 Intended use template 222
  17.3.2 Risk management file template 224
  17.3.3 Software development plan template 224
  17.3.4 Software requirements specification template 225
  17.3.5 Software architectural design template 226
  17.3.6 Software detailed design template 227
  17.3.7 Test plan template 228
  17.3.8 Test case specification template 229
  17.3.9 Test procedure specification template 229
  17.3.10 Test incident report template 230
  17.3.11 Test summary report template 231
  17.3.12 Review report template 231
  17.3.13 Meeting report template 231

18 Test management and defect management 233
18.1 Software testing principles 233
18.2 Software testing strategies 234
18.3 A software testing process
   18.3.1 Test planning, monitoring, and control
   18.3.2 Test analysis
   18.3.3 Test design
   18.3.4 Test implementation
   18.3.5 Test execution
   18.3.6 Test evaluation exit criteria
   18.3.7 Test closure
18.4 Test metrics
18.5 Defect management

19 Change management, configuration management, and change management
   19.1 Change management
   19.2 Configuration management
   19.3 Incident management

PART VI Conclusions

20 Conclusions
   20.1 Perspectives
   20.2 Criticality
   20.3 Conclusions

References
Index