



POLARIS U.K. LIMITED

**Planning for Electronic Trading
Implementation Using
ProductWriter**

Version: 1.3

Updated: 8th August 2006

Contents

DOCUMENT OBJECTIVES	4
MARKET ANALYSIS	5
ASSESS MARKET (R).....	5
BRIEF SALESFORCE (R).....	5
OBTAIN OTHER MARKET INFORMATION (R).....	5
WRITE ANALYSIS (R).....	5
OPPORTUNITIES ANALYSIS AND SCENARIO DEVELOPMENT	6
HIGH LEVEL PROCESS ANALYSIS (I).....	6
DEVELOP COST AND CYCLE TIME SIMULATIONS (I).....	6
DEVELOP SCENARIOS FOR NEW PROCESSES (I).....	6
MODEL AND SIMULATE ALTERNATIVE OPTIONS (I).....	6
SELECT TARGET DEVELOPMENT PROCESSES (I).....	6
COST BENEFIT ANALYSIS AND FEASIBILITY STUDY	7
SET TARGETS (I).....	7
CARRY OUT ANALYSIS (I).....	7
PREPARE BUSINESS CASE (I).....	7
BUSINESS ANALYSIS	8
CONSIDER PRODUCT DEVELOPMENT IMPLICATIONS (R).....	8
OUTSOURCING VERSUS IN-HOUSE (R).....	8
MARKETING STRATEGY	9
DETERMINE TIMINGS (R).....	9
DETERMINE SUCCESS CHANCE/RISK (R).....	9
SET SERVICE STANDARDS (I).....	9
DISTRIBUTION CHANNELS (I).....	9
AGREE PLANS WITH TRADING PARTNERS (I).....	9
SOFTWARE HOUSE PANELS (I).....	9
PROJECT PLANNING	10
GENERAL.....	10
DEFINE OUTLINE PROJECT OBJECTIVES (I).....	10
AUTHORISE PROJECT COMMENCEMENT (I).....	10
INVESTIGATE TECHNICAL IMPLEMENTATION ISSUES (I).....	10
RESOURCE PLANNING	11
SKILLSET (I).....	11
AVAILABILITY (I).....	11
PRODUCT SELECTION AND SPECIFICATION	12
INVESTIGATE CURRENT PRODUCT PORTFOLIO (R).....	12
IDENTIFY BASE INSURANCE PRODUCT TO BE USED (R).....	12
REVIEW WAY OF WORKING (R).....	12
REVIEW PRODUCT SPECIFICATION FROM AN UNDERWRITING PERSPECTIVE (R).....	12
ENSURE PRODUCT SPECIFICATION IS ACCURATE AND COMPLETE (R).....	13
SIGN OFF SPECIFICATION (R).....	13
IMPLEMENTING PRODUCTWRITER®	14
ARRANGE PURCHASE OF PRODUCTWRITER® AND DATA DICTIONARY (I).....	14
CONSIDER/ATTEND TRAINING (I).....	14
UNDERSTAND DATA DICTIONARY (I).....	14

REVIEW PRODUCT SPECIFICATION, POLARIS DATA DICTIONARY (R).....	14
REVIEW THE DATA GATHERING/CONVERSION IMPLICATIONS FOR EXISTING BUSINESS (R).....	14
TECHNICAL ANALYSIS.....	16
PRODUCTWRITER [®] RULES (R).....	16
DATA DICTIONARY RESTRICTIONS (R).....	16
CODE LISTS (I).....	16
TABLES AND GROUPS (R).....	16
PLAN TESTING (R).....	16
DOCUMENT SPECIFICATION AND QUALITY CONTROL (I).....	16
SYSTEMS DESIGN.....	18
SPECIFY THE PROCESSING FLOW (I).....	18
FUNCTIONS CARRIED OUT BY INSURER SUPPLIED COMPONENTS (I).....	18
CONFIRM DELIVERY OF REQUIRED SYSTEM COMPONENTS (I).....	18
PREPARATION FOR TECHNICAL ACTIVITY.....	19
OBTAIN HARDWARE AND INSTALL SOFTWARE (I).....	19
ASSIGN PRODUCTWRITER [®] RESOURCE (BUSINESS AND IT, AS APPLICABLE) (I).....	19
ATTEND FURTHER TRAINING (R).....	19
PLAN POS COMPUTER SYSTEMS PROJECT (I).....	19
BUILD PRODUCTWRITER[®] INSURER PRODUCT.....	20
GENERAL.....	20
PLAN USE OF INPUT RATING FROM THE DATA DICTIONARY (R).....	20
PLAN YOUR DESIRED OUTPUT RESULTS (R).....	20
PLAN THE STRUCTURE OF RULES PROCESSING (R).....	20
BUILD THE INSURANCE PRODUCT USING PRODUCTWRITER [®] (R).....	20
TESTING.....	21
CREATE TEST DATA (R).....	21
TEST INSURANCE PRODUCT THROUGH PRODUCTWRITER [®] (R).....	21
TEST PACKS (R).....	21
TESTING RESOURCES (R).....	21
ACCREDITATION.....	21
SIGN OFF TEST AND PRODUCT BUILD (R).....	22
GENERAL.....	23
USER SUPPORT (R).....	23
TRAINING PLANS (R).....	23
END TO END TESTING (R).....	23
POST IMPLEMENTATION CONSIDERATIONS (R).....	24
APPENDIX A.....	25
ADDITIONAL TASK LIST.....	25

Document Objectives

This document identifies and describes suggested tasks and events that should occur during the implementation of ProductWriter[®] as well as the process of building and distributing a product(s) using the Polaris Toolset. Collectively the list of tasks are not applicable to just one individual and can be divided up amongst a number of different resources in different organisations eg. IT managers, programmers, underwriters.

The main body of the document describes tasks relevant to the insurer while an additional list of tasks, including the Run Time Environment (RTE) implementation, to be carried out by software houses, insurers and Polaris is included in Appendix A. However, the entire list should be intelligible to all parties and members of each “team” should be aware of tasks that other teams within other organisations will have to perform.

Obviously, some of the tasks described will only occur once - ie. during the implementation of the ProductWriter[®] or RTE software - whereas other tasks should recur every time a new product is developed. For the purposes of this document the initial tasks have been qualified with the letter “I” while the recurring tasks have been marked with an “R”.

The overall project constitutes suggested tasks based upon previous experience as well as the contributions of software houses and insurers and should not be interpreted as a specific recommendation or direction. Each implementation project is unique and will encounter local technical and management issues. However, the list acts as a very useful starting point and template for overall project management.

Readers may find that not all tasks will be specifically relevant to them during the definition and building of products. This may be the case; however, there are a number of different ways that the development of ProductWriter[®] defined products can take place - eg. the insurer may choose to develop inhouse or outsource to a third party software house; the document attempts to identify the different scenarios and provide a useful checklist for all of them.

The reader should ensure that inhouse project management procedures and selected development methodologies are considered carefully alongside the information supplied below.

Market Analysis

Assess market (R)

Gather as much information as possible to help assess the opportunities for business development and retention. “Niche” products offer competitive advantage. Opportunities will exist for new niche products and underwriters will normally evaluate the viability of the introduction of new products. Improving existing products will often provide a quicker solution as introducing new products and new underwriting factors can slow down the estimated lead time due to the additional requirement for back-office changes.

In any case speed to market will be vital. Marketing opportunities are generally triggered by underwriters.

Brief salesforce (R)

The salesforce should be aware of how to market products in view of impending changes. An up-to-date knowledge of new products should be maintained so that the salesforce is able to promote new products rapidly.

Obtain other market information (R)

It is important to keep a watchful eye on the rest of the market in order to establish what other products are on offer and how they are being distributed. Niche products can gain particular competitive advantage and it is important to establish what other niche products exist. Feedback can be obtained from the relevant software house regarding product performance.

Write analysis (R)

Work and procedures carried out at this stage should be maintained in a written form which makes it suitable for revision and reference. As will be seen throughout this project, coherent documentation is of the utmost importance in establishing consistency, minimising unnecessary repetition and avoiding problems when, for example, employees leave the company.

Opportunities Analysis and Scenario Development

High level process analysis (I)

In order to establish and maintain a robust infrastructure incorporating insurers, software houses and intermediaries, it is important to understand all the processes which take place and the relationships that exist at various stages. Before analysing the specific detailed processes it is important to have an understanding of the high level processes that occur both internally and with trading partner(s).

Develop cost and cycle time simulations (I)

Cost and cycle time analysis is key within the processes that exist between organisations. Simulations can be built to show the current “state”. Cost and cycle time analysis not only affects improvement in general operational processes but could also affect the decision as to whether to introduce a new product. Often the negative cost effectiveness of a new product is only realised years later.

Develop scenarios for new processes (I)

The effects of implementing new processes need to be analysed. If the procedures that currently exist - for example with the delivery of new products to software houses - are inadequate in terms of cost and cycle time, these should be improved.

Model and simulate alternative options (I)

Simulations can be used to demonstrate the effects of a number of different variables eg. resources available, work in progress, performance. There are a number of packages which exist which can assist with modelling existing procedures and alternative options. When building new products, templates can be used to assess viability of a particular niche product.

Select target development processes (I)

The key processes which need development in light of the output from the simulations should be approached first.

Cost benefit analysis and feasibility study

Set targets (I)

The financial implications of implementing ProductWriter[®] - generically known as cost benefit analysis. Many projects are undertaken as acts of faith with the benefits unknown. It is important to identify and set required targets in order to achieve quantifiable conclusions.

Carry out analysis (I)

Cost benefit analysis is an important step. Various methodologies for and approaches to cost benefit analysis already exist¹.

Prepare business case (I)

This should integrate the various elements of the analysis into a clear statement of the nature of the investment and the project returns.

¹ An extensive generic guide to cost benefit analysis has been published and is available from Polaris.

Business Analysis

Consider product development implications (R)

Product development relates to two different areas: a) products which already exist but are suitable for definition using ProductWriter[®] - benefits can be realised using the cost and cycle time analyses above; b) new products which are suitable for definition using ProductWriter[®] or which, in some cases, may have been invoked directly by the implementation of ProductWriter[®]. Whichever category a product falls into, it should be easily implementable using ProductWriter[®].

It is important to consider whether all software house systems can support the product or whether variations will be needed for different software houses to cope with the limitations of the data they capture. The planning process must allow time for this work to be completed.

Subsequent impact on back-office systems should be considered.

Outsourcing versus in-house (R)

Consider benefits and disadvantages of outsourcing or in-house production. Some tasks may have to be outsourced or will be subject to external consultancy. Outsourced organisations will already have experience of running similar projects on a similar scale. They will have the experience of using structured methodologies and will possess resources with the right skillsets to perform the necessary tasks.

However, a dedicated internal team with the right skillset can ensure continuity during the development period; this approach can also help to avoid any delays associated with physically passing a specification for each product to a third party and improves the maintainability of all products. If in-house production is to take place, a good understanding of development methodologies should be gained.

If outsourcing, a strong alliance and trust should be maintained with the outsourced business partner.

Marketing Strategy

Determine timings (R)

The timing for the implementation of ProductWriter[®] itself will probably be governed by a strategic need to increase the speed to market of products. The timing of the release of individual products also needs to be addressed. What other products are being marketed by the insurer, other competitors and when are they being marketed.

Determine success chance/risk (R)

It is important to keep a close eye on the market place to ensure that the competition is identified. Consider success factors for the initial ProductWriter[®] implementation and the success factors for each product.

Set service standards (I)

The implementation of new systems should infer an improvement in service standards and these should be recognised and documented.

Distribution channels (I)

Analyse impact on other, perhaps more traditional, distribution channels.

Agree plans with trading partners (I)

These should be signed off formally so that no misunderstandings emerge over responsibilities.

Software house panels (I)

Select membership of appropriate broker software house panels. When joining a software house panel, all relevant charges should be considered and contracts reviewed and signed. Service levels with the broker software house should be considered - primarily service levels for:

1. The introduction of new products
2. Data dictionary changes (including the support/withdrawal of products based on back versions)
3. The introduction of rate changes.

Project Planning

General

Communication and coordination is key during project planning and subsequent stages. Ongoing reviews should take place during the development phases. If necessary, it may be necessary to physically dovetail plans with those of trading partners depending on the closeness of the relationship.

Define outline project objectives (I)

Describe deliverables, dates, resources, POS activity targets, responsibilities and technical architecture. Initial implementation of ProductWriter[®] will obviously require the greater amount of time and investment. When further product definition takes place using ProductWriter[®] a structured, generic plan should be adhered to each time.

Authorise project commencement (I)

Appoint project manager. He/she should have the appropriate authority to make decisions on both an underwriting and technical level. The relevant financial outlay should have been authorised.

Investigate technical implementation issues (I)

Initial analysis of hardware and resourcing issues to ensure feasibility of implementation of ProductWriter[®] should take place at an early stage. This includes analysis of insurer back-office systems and EDI requirements.

Resource Planning

Skillset (I)

Over the total project a variety of skillsets will be required ranging from marketing through IT and these need to be identified and provided.

For the actual product development element it is essential to ensure the right skillset is selected. It is necessary to look for hybrid skills - a mixture of underwriting and IT experience as outlined below.

Task	Resource
Product specification	Underwriter
Rule writing in ProductWriter®	IT or underwriter with programming aptitude and a good understanding of the ProductWriter® data dictionary
Testing	Underwriter but from an alternative source

It is also advisable to set up an internal ProductWriter® user group with those proficient in the use of ProductWriter® and to participate in external user groups.

Availability (I)

Ensure that the workforce will be available at necessary times throughout the project. Overall team productivity will also be ensured by maintaining a consistent workforce.

Product selection and specification

Investigate current product portfolio (R)

Analyse rating, referral and underwriting of current portfolio. A suitable market opportunity should already have been realised for a new niche product.

Identify base insurance product to be used (R)

Select product for implementation. The product should lend itself to implementation using ProductWriter[®]. The following elements should be considered as a priority for each product to be specified using ProductWriter[®] - note, the whole business cycle should be considered:

- Design and technical context
- Cover
- Pricing. Target premium
- Simplification of the rating basis
- Documentation - including technical wordings and prospectus
- Marketing support
- Legal issues.

Review way of working (R)

Detailed discussions should take place with software houses and brokers. This should include debating how to handle not only new business but also:

- Mid term amendments
- Renewals - will this be led by the insurer or the broker?
- Documentation - where will this be produced?

Review product specification from an underwriting perspective (R)

Revise underwriting and rating specification to meet the autorating and reduced referral targets set. A Rapid Application Development (RAD) approach can also be adopted here. For example, it may be difficult to achieve the minimum referral targets from day one. Rules may have to be written at a later date in order to fill the “holes” in the scheme that formally invoked a referral.

The product specification also needs to be considered from a broker perspective. What does the broker system provide in terms of information? Consider the impact of changes on the project.

Ensure product specification is accurate and complete (R)

Produce a single product specification document for each product. Although this document endorses a RAD approach to product definition, it is important for underwriters to have a clear understanding of what is to be built before building it. Building a product in ProductWriter[®] from scratch, without a clear detailed paper definition² and plan, is not recommended. ‘Contents of a Product Definition’ (hyperlink) provides a more detailed analysis of what to include in the product definition for commercial lines best practices. Much of this information is also pertinent to the definition of personal lines products.

Many products have similar characteristics. Templates should therefore be built using the base level underwriting criteria with individual product anomalies written on top of these. Where multiple, similar products are being written this approach will offer substantial savings in time and cost³. Obviously this also implies a considerable amount of pre-planning.

Sign off specification (R)

Specification documentation should be formally signed off.

² Polaris have example “requirement definition” and “technical specification” documents which accompany the “skeleton schemes”. [HYPERLINKS](#)

³ Polaris has developed templates or “skeleton schemes” which are available to ProductWriter[®] users.

Implementing ProductWriter®

Arrange purchase of ProductWriter® and data dictionary (I)

Commence pre-sales discussions, arrange finances and discuss contractual agreements with Polaris. Signing of confidentiality agreements may be deemed necessary by either party at this stage. This is also a good opportunity for both companies to identify respective contacts and to agree any consultancy - Polaris advises a minimum amount of initial consultancy. At this stage it is important to evaluate what solutions ProductWriter® offers and what it does not offer.

Consider/attend training (I)

Availability of course placements and staff to be trained should be considered. Early training will also give a broader technical understanding of the software, the data dictionaries and potential issues that might arise during implementation. Note that expertise gained through attending the ProductWriter® course should be made use of as soon as possible. Also, as a rule, only individuals who have been trained in ProductWriter® should participate in consultancy supplied by Polaris.

It is advisable to send more than one delegate on a training course. A combination of at least one underwriter and one member of IT is ideal.

Understand data dictionary (I)

Ensure that the data dictionary is fully understood, otherwise build time may be increased and performance impaired at a later date.

Review product specification, Polaris data dictionary (R)

Review product specification against the Polaris data dictionary. Ensure the product can be built using the data dictionary. Investigate in detail the implications of change from free text descriptions of rating factors to list based selection. Identify any non-supported requirements and ensure these are discussed with Polaris at an early opportunity and that they can be incorporated into the data dictionary in the required timeframe.

Review the data gathering/conversion implications for existing business (R)

Ensure existing policies can be converted to the new representation and that any missing data is captured. Data capture is not only an issue relating to ProductWriter® but may also relate to:

- limits of the back-office system
- broker software system limits.

Discussions need to take place with the software house concerning data capture limitations. Does the broker system capture the required data, can it capture it? Investigate how specific systems cope with data capture.⁴

⁴ See also Appendix A, 4) Install the insurer ProductWriter® product into the POS system

Technical analysis

ProductWriter® rules (R)

Consider approach to writing ProductWriter® rules. Consider how rules will perform calculations etc. The ProductWriter® software includes a test harness which should be used continuously throughout the build process. A sensible test pack should be developed for each product. Generic/template test packs could also be used.

Each rule performs a particular function and may be of use across several schemes. It is suggested, therefore, that a structured approach is adopted. For example, a library of rule reports could be built up - rules can be dated, versioned, described and modifications documented. This greatly facilitates rule maintenance.

Data dictionary restrictions (R)

Are there any restrictions in the current data dictionary? Following an initial analysis, these should be notified as soon as possible to the relevant departments in order to minimise delay at a later date.

Code lists (I)

Consider issue of code list change requests. New codes may need to be added and these should be notified to the relevant parties - eg. Polaris, ABI.

Tables and groups (R)

Identify relevant tables and groups for maintenance. The maintenance overhead for tables and groups within schemes should not be underestimated. Each product will be subject to a number of changes during its life time to reflect market forces and, as the number of products defined rises, so the need for maintenance will increase.

A systematic and controlled approach to table and group maintenance should be adopted as well as possible version control.

Plan testing (R)

A realistic amount of time should be set aside for testing. It is often easy to underestimate this time period.

Document specification and quality control (I)

It is important that much of the work carried out to date is documented and summarised in order to provide a template for future work and analysis. This requires a degree of quality control.

Systems Design

Specify the processing flow (I)

Specify the processing flow (based upon the current systems). Define the POS processing and communications.

Functions carried out by insurer supplied components (I)

Specify in detail the functions carried out by the insurer supplied ProductWriter[®] components within the processing flow. Specify what rule sets, forms and EDI messages will be built and what functions they will provide.

Confirm delivery of required system components (I)

Confirm the capability of ProductWriter[®] to deliver the required system components. It has been mentioned that products should be suitable for implementation using ProductWriter[®]. In the same way, it must be proved that ProductWriter[®] is able to deliver improvements in product definition over existing processes. Consider which functions ProductWriter[®] is required to offer and which functions will be carried out outside ProductWriter[®]. For example, EDI functionality is provided within the Polaris scope by a combination of ProductWriter[®] and a third party supplied gateway (Perwill) but it may be preferable to handle it using other, pre-existing, facilities if these are available.

Preparation for technical activity

Obtain hardware and install software (I)

Discuss costs and licensing with Polaris. Define delivery dates. Bear in mind that legal departments often require a substantial amount of time for reviewing agreements.

Acquire PC and arrange installation of the software and relevant data dictionaries. This can be achieved by either sending a PC for installation and configuration at Polaris or arranging an on-site installation by Polaris technical staff. In the past it has been proven beneficial to install a copy of the software on a laptop for portability and to enable physical access by various members of a group.

Assign ProductWriter[®] resource (business and IT, as applicable) (I)

Assign internal or buy external resources and assistance to build and test the ProductWriter[®] product specification⁵. This may involve completely outsourcing the technical activities of product building or ensuring that the relevant, trained skills are available in-house. Contractors may have to be brought in on a technical level or on a business level.

Attend further training (R)

Now that an initial skillset has been developed, further individuals may need to undergo training. Determine the availability of course placements and staff to be trained. Note that expertise gained through attending the PW course should be made use of as soon as possible.

Plan POS computer systems project (I)

Arrange and utilise implementation consultancy. Prepare and carry out the project plan; discuss and agree with all parties internally and externally.

⁵ See Outsourcing versus in-house

Build ProductWriter® insurer product

General

In all cases it is important to consider declines, refers, quotes, new business, mid-term amendments, renewals.

Plan use of input rating from the data dictionary (R)

Have a repeatable and logical data requirement. Continue analysis of existing documentation for existing products. Access to rates and tables and translation of rates and tables.

Plan your desired output results (R)

Have a repeatable and logical production of results.

Plan the structure of rules processing (R)

Have logical units of processing which can be tested and is easily understood. When building any product it is important to consider ultimately the maintainability of the ProductWriter® rule files. As mentioned earlier a structured, report based, reference library of rules should already have been developed.

Build the insurance product using ProductWriter® (R)

Load group data, tables and write rules. The pre-planning which has already taken place should enable the product to be easily built and, as technical understanding of the software improves, it should be possible to minimise the actual product build time to the utmost.⁶

⁶ See also Appendix A; 4) Install the insurer ProductWriter® product into the POS system, 7) Develop the back-office application

Testing

Create test data (R)

Employ testing methodologies similar to testing other software systems to ensure the logic of the 'rules' is correct, complete and gives the right answers in terms of premium and other outputs. Select a test team which has not been hitherto directly involved with writing insurance products using ProductWriter[®]. However, testers should have adequate underwriting experience and IT experience to understand the complexities of building products. The amount of time given over to testing in the overall project plan should not be underestimated.

Test insurance product through ProductWriter[®] (R)

The ProductWriter[®] toolset provides three testing tools:

- ◆ a test harness which can be used for testing schemes during and subsequent to the development stage
- ◆ a debugger which allows schemes rules to be checked sequentially for integrity
- ◆ a batch tester which enables test data to be created using standard spreadsheet facilities and then run to compare expected against actual results.

This testing in the long run will not be adequate alone. Further testing mechanisms should be planned and implemented in conjunction with the broking software house and/or each trading partner.

Test packs (R)

Use scheme test packs. Test packs should be created for each scheme developed and should comprise test data and expected results. Due to the similarity of some products, test packs can be re-used and modified.

Testing resources (R)

Consider skill-set/experience and availability of testers.

Accreditation

Polaris offers an accreditation mechanism designed to test EDI message generation. This tests and endorses the fact that the EDI message has been correctly populated with pre-specified test data. It should be considered as a final quality check once all in-house testing is complete.

Sign off test and product build (R)

The development of each product should have a review and sign off date in order to achieve realistic project timescales.

General

User support (R)

Plan what support is required for new users including documentation and what support is required for the new product.

Training plans (R)

Decide who needs training and how this will be provided on an ongoing basis. Investigate availability of training courses. Train the relevant staff. Train trainers⁷.

End to end testing (R)

Full acceptance test of the total business cycle between partners. From the ProductWriter[®] point of view, the best way to check the total integrity and reliability of the product written is to test on an end-to-end basis. This implies a suitable relationship with a software house partner and involvement of the broker system in the test process.

The entire chain from broker system to insurer back-office system including account reconciliation ultimately needs to be tested and the insurer should decide how much of this is achievable. Polaris strongly recommends end-to-end testing.

The following test areas need to be considered:

1. Insurer back-office system
2. Broker system and broker acceptance testing - this may often take place as part of the final test phase
3. Accounts
4. Auditing
5. EDI testing - incomplete products may not fully populate the message and the message/gateway may fail. These discrepancies may not be evident until the EDI message is built.

These factors should be considered in a full cycle context.⁸

⁷ See Consider/Attend Training.

⁸ See also Appendix A; 3) RTE Testing, 6) Develop and test the POS EDI capability, 7) Develop the back-office application.

Post implementation considerations (R)

At all points during the development consideration should have been given to the need for future ease of maintenance. Once the development is completed it will be necessary to have the necessary support infrastructure in place to consider and apply amendments as necessary which might arise for areas such as amendments to the Polaris dictionary or software, or from software house requirements.

Following completion of the product a full post implementation review should be conducted to review all aspects of the project and any lessons learnt should be applied to subsequent product builds.

Appendix A

Additional Task List

The following is a further list of tasks which should be carried out by ALL parties - insurers, software houses and Polaris.

Task	Description	Responsibility
1) POS system design		
Consider POS system design	Data capture, screen flow and data storage. Much of the POS system design will be tested at the end-to-end testing stage. What functions are supported?	I,S
Consider Polaris RTE integration	Discuss integration issues regarding the ProductWriter [®] Run Time Environment (RTE) with Polaris. Consider Polaris RTE data handling and process invocation.	S,P
Specify technical implementation environment	Including front-end application, data storage and communications.	I,S
2) Develop the POS application		
Review and agree hardware platform	A review and agreement of the relevant hardware platform should take place with Polaris.	S
Design POS screens and flow	Using insurer product and Polaris data dictionary.	I,S
Data stores and data mapping	Design POS data stores and data mapping to existing system files.	S
Design POS system administration	Access control, audit, recovery, authority, overrides,	I,S

requirements	data dictionary and insurer product maintenance control procedures.	
Add Polaris RTE use to application design	The fact that the RTE will be used should be incorporated into the application design.	S
Receive Polaris RTE software and development briefing	Arrange delivery of RTE software and sign licence agreement. Bear in mind that legal departments often require a substantial amount of time for reviewing agreements. Arrange and utilise consultancy relating to RTE implementation.	S,P
Build POS software	Build POS software including system administration processes. Build screens, data stores, admin.	S
3) RTE testing		
Create system test data	Test data should be created which tests a number of different areas and eventualities. Test data should be created by individuals with underwriting experience as well as IT experience.	I,S
Acquire, test data dictionary and test insurer product	Polaris can supply a test dictionary and product.	S,P
Test application interfaces	Test application interfaces and system administration processes. Test screens, data stores and invocation and system administration processes.	I,S
Testing resources	Consider skill-set/experience and availability of testers.	I,S
4) Install the insurer		

ProductWriter[®] product into POS system		
Obtain and install current data dictionary from Polaris	Users will be automatically provided with updates to the data dictionaries as part of Polaris support service levels. Users should ensure that they have the latest version of the dictionary installed in order to avoid support problems and to comply with the terms of Polaris licensing. Users also need to consider what level the software house is currently supporting.	I,S,P
Obtain and install insurer product and test data	The final arrangements should be made for receiving the (first) insurer products and test data. Ensure that all relevant contracts have been signed.	I,S,P
Test insurer product through POS system	See end-to-end testing.	S,P
5) Acceptance test the POS system		
Acceptance testing of POS system	Broker user testing should take place. This can take place in isolation but ultimately forms part of the overall end-to-end testing.	I,S
Polaris RTE Population Testing	Polaris tests the loading, unloading and mapping of data through the data dictionary from both screens and data stores in the POS application. Establishing a ground point for implementation and product testing.	I,S,P
Testing resources	Consider skill-set/experience and availability of testers.	I,S
6) Develop and test the POS EDI		

capability		
Receive EDI message specification	<p>Obtain message specification - Motor and Household messages from the ABI and Commercial message from Polaris.</p> <p>Obtain the message to dictionary mapping specification from Polaris.</p> <p>Some degree of familiarity with EDI standards and mapping might prove useful.</p>	S,P
Analyse message specification and produce impact analysis	<p>Look for changes to front-office data stores and gateways. Ensure that message sent can be accepted and processed.</p>	S
Specify EDI integration requirements, including audit and control	<p>Document the degree of integration between the EDI translator and the broking application including what audit/control facilities are needed and how these will be provided. If using the Perwill EDI gateway software, Perwill can assist with integration advice and support.</p>	S
Review EDI mapping specification	<p>If using Perwill EDI software Polaris will have predefined a standard mapping from/to the dictionary properties. Check whether this needs to be tailored to meet specific requirements.</p> <p>If using other EDI software the mapping will need to be created and tested.</p>	S
Acquire EDI hardware and communications software	<p>Polaris offer a free licence for member software houses to use the Perwill EDI gateway. Other gateway products may be used if preferred.</p>	S
Install communications hardware and software	<p>Install communications hardware and software and set up message handling if necessary. Perwill EDI software includes a communication module.</p>	S

Test EDI mapping	Use test data to ensure the EDI integration is correct.	S
Test communications hardware and software	Carry out supplier recommended tests to ensure communication links can be established with trading partners	S
Test EDI message generation and processing	Verify the broking application, incorporating the RTE and the EDI software, is generating the required message content and that the communications software is transmitting this correctly.	S
Test non-ProductWriter[®] EDI message handling (if relevant)	Verify applications and gateways are working correctly for other messages.	S
Accreditation with Polaris	Arrange date for accreditation check with Polaris. Send Polaris sample message plus test data used to generate it.	S,P
Joint insurer/broker message testing	Joint insurer / broker message testing.	I,S
Sign Interchange Agreement	Polaris provides a standard interchange agreement which specifies terms which should be adhered to by trading partners when exchanging electronic messages. It may be considered necessary to sign this agreement.	I,S
7) Develop the back-office application		
Analyse message specification and produce impact analysis	Look for changes to back-office data stores and gateways. Ensure that messages can be accepted and processed.	I
Design back-office EDI system administration requirements	Access control, audit, recovery.	I
Design back-office validation	See introduction note 7.	I

procedures		
Develop and test data storage capability	Review existing internal data store - databases and files to ensure complete storage of message content.	I
Modify and test communication gateway capability	Modify gateway, etc. if required to support message format and processing and for sending and receiving messages.	I
Test gateway processing.	Test gateway processing of received messages and EDI messages to be sent.	I,P
Joint insurer/broker message testing	See end-to-end testing.	I,S

I = Insurer
S = Software House
P = Polaris