

**Table 1: Comparison of Envision and TPG3 Concepts.**

Envision	TPG3
<b>Exec:</b> abstract base class for <i>Test</i> , <i>Subflow</i> , and <i>Bin</i> .	<b>Test:</b> abstract base class of <i>TestBlock</i> and <i>TestFlow</i> , and specific test methods (e.g. <i>VirTestFn</i> ); contains id, pre/body/post processing instructions. The body represents the test method associated with the particular test type, and its parameters.
<b>Flow:</b> multiple entry points.	<b>TestFlow:</b> single entry point (same as Envision OnStart); derived from <i>TestBlock</i> : <i>TestFlow</i> is the top level <i>TestBlock</i> .
<b>Subflow:</b> graph of <i>Nodes</i> .	<b>TestBlock:</b> graph of <i>FlowNodes</i> ; base class of <i>TestFlow</i> .
<b>Node:</b> contains id, pointer to <i>Exec</i> instance, Calibration, Spec Block, and Category.	<b>FlowNode:</b> contains id and potentially pre/body/post processing instructions. Body is a pointer to a <i>Test</i> instance and the only required processing instruction.
<b>TestMethod:</b> contains arguments.	<b>TestFn:</b> contains arguments. Base class for virtual and target functional tests. One such base class exists for each test method.
	<b>VirTestFn:</b> virtual functional test. Derived from <i>Test</i> and <i>TestFn</i> .
<b>BinMap:</b> maps software to hardware bins. Single dimensional ?	<b>BinMap:</b> maps software to hardware bins. N dimensional.
<b>Bin:</b> contains name, derived from <i>Exec</i> .	<b>Software Bin:</b> optional action associated with a Test post process.